



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 2

SISTEMA DE GESTIÓN DE INFORMACIÓN PARA EL VICEDECANATO DE INVESTIGACIÓN Y POSTGRADO DE LA FACULTAD 2

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores: Osmar Capote Vázquez

Raúl Vázquez Leyva

Tutor: MSc. Madelis Pérez Gil

La Habana, 2015

*"Cualquiera puede hacer complicado algo simple. La creatividad consiste en
hacer simple lo complicado"*
Charles Mingus

Osmar

A mis padres quienes han sido siempre mi motor impulsor para lograr cada objetivo que me he propuesto siendo este el más grande hasta ahora ser un Ingeniero. Mis padres han estado a mi lado en todo momento, en ocasiones de tristeza, de nerviosismos y de alegría como el que estoy viviendo actualmente. Por ustedes he llegado hasta aquí porque ustedes son las pupilas de mis ojos, ustedes son la razón por la cual existo LOS AMO muchísimo. Y doy las gracias a DIOS por yo ser el fruto de unos padres tan maravillosos, por haber tenido en cinco años compañeros de aula inolvidables, por haber recibido clases de un buen claustro profesoral, amistades que conocí en el transcurso de la vida universitaria y por haber conocido a la mujer de mi vida.

No podía seguir escribiendo sin dedicarle luego de mis padres a la persona que me formó como debe de formarse todo un profesional de las Ciencias Informáticas mi novia, guía, mi ayuda idónea, mi tutora, mi futura esposa y madre de mis hijos Madelis Pérez Gil. MAGUI como te lo prometí en uno de los muchos días que platicábamos que MIVCA te iba a defraudar y esta tesis es un ejemplo de ello, quiero que sepas que he llegado hasta aquí y he logrado cumplir mi mayor sueño todo gracias a tu apoyo incondicional e intachable, gracias por tu persistencia, paciencia, consejos, por tus sugerencias, por tu fe en mí, por todo esto y muchísimo más que tengo que agradecerte que no bastaría un documento de 80 páginas quiero que sepas que eres fundamental en mi vida, que te amo y TE NECESITO.

A mi abuela Adanela por ser esa persona que de su boca nunca ha salido un NO cuando se trata de mí, abuela te quiero mucho y te tengo como mi segunda madre, eres muy especial para mí.

A mi abuelo Misnardo el cual no hay palabras para describir lo importante que es en mi vida como dice un compañero mío YO LO DESCRIBO COMO ALGO INDESCRIPIBLE.

A mi abuela María la cual siempre ha creído en mí y me ha inspirado como ejemplo a seguir por su fortaleza, amor y esperanza. Mi viejita gracias por existir en mi vida.

A mi abuelo Isidro el cual a pesar de ser una persona de pocas palabras jamás se me va a olvidar lo que me dijo el 27 de diciembre del 2014 en el balcón de su casa Osmarito tú eres los ojos de tus padres y porque lo sabes se que nunca los vas a decepcionar.

A toda mi familia por ser de una forma u otra parte de este gran logro que es ser un Ingeniero de las Ciencias Informáticas.

Raúl

Al Rey de Reyes y señor de señores, mediante el cual podemos hacer todas las cosas, a mi familia y amigos, a mis hermanos en la fe.

Osmar

Le agradezco a Ángel Luis porque sin duda alguna no habría terminado esta tesis sin su ayuda, a Hassán por enseñarme a escribir un documento con la calidad que en verdad requiere, a Eblis por ser esa persona que ha pesar de habernos conocido en este último año se comportó conmigo como un verdadero amigo que es para mí. A Yayi por haberme asustado esa noche en el apto, a madelis por ser tan comprensiva, tan excelente guía y maestra. A mis padres quienes prácticamente vivieron conmigo cada momento de esta tesis, gracias por ser tan fieles y confiar ciegamente en mí y a DIOS por permitirme conocer a personas tan especiales en mi vida. A todos gracias por haberme guiado por el camino hacia el éxito.

Raúl

Le agradezco a Dios sobre todas las cosas, porque solo él es dador de la vida, del conocimiento, de la sabiduría y de la palabra de fuerza y aliento cuando creemos que no podemos, por su fidelidad en todo tiempo, y por enseñarme con esta tesis que él es más grande que cualquier mano humana. A mi extensa familia, mi mamá, papá, madrastra, padrastro, abuelos, primos y tíos, factor incondicional en mi vida profesional. A los que contribuyeron aquí en la habana a mi vida durante estos cinco años. A mis hermanos amados en Cristo Jesús, por ser mi familia UCI, por estar siempre en el momento preciso con el consejo necesario.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Osmar Capote Vázquez
Autor

Raúl Vázquez Leyva
Autor

MSc. Madelis Pérez Gil
Tutor

La informatización de procesos manuales o semi-informatizados, es una de las tareas que más ha evolucionado el entorno a finales del siglo XX y lo que va del XXI. Para llevar a cabo dicha informatización en Cuba se le ha dado a la Universidad de las Ciencias Informáticas (UCI) la importante tarea de desarrollar la mayoría de las actividades en este sentido. La universidad cuenta con un personal docente en su mayoría graduados de la institución. A estos se les da la posibilidad de continuar su formación post-graduada en el centro. En cada facultad de la UCI se encuentra el área del Vicedecanato de Investigación y Postgrado (VIP) que se encarga de llevar la planificación y control del desarrollo científico-investigativo de sus profesores y especialistas. En el área del VIP aún existen flujos de datos manuales o semi-informatizados, y se detectan un grupo de necesidades específicas de la facultad que ningún sistema implementa. Por lo que se hace necesario facilitar el proceso de Gestión de Información. La actual investigación pretende dar una solución práctica, pues se desarrolló un sistema informático que automatiza el proceso de Gestión de Información referente al área del Vicedecanato de Investigación y Postgrado VIP de la Facultad 2.

Palabras clave: gestión, información, investigación, postgrado, sistema.

Introducción	1
1 Fundamentación Teórica	8
1.1 Introducción	8
1.2 Conceptos Asociados	8
1.2.1 Gestión de Información	8
1.2.2 Sistema de Gestión de Información Sistema de Gestión de Información (SGI)	9
1.2.3 Sistema Informático de Gestión de Información	9
1.3 Estudio del Estado del Arte	9
1.3.1 Internacional	9
1.3.2 (UCI)	11
1.4 Conclusiones del estudio	13
1.5 Metodologías de Desarrollo de Software	14
1.5.1 Programación Extrema Programación Extrema (XP)	14
1.6 Herramientas y lenguajes informáticos	15
1.6.1 Lenguaje de Marcado de Hipertexto(HTML) 5	15
1.6.2 Hojas de estilo en cascada(CSS) 3	16
1.6.3 JavaScript	16
1.6.4 Hypertext Pre-Processor (PHP) 5	17
1.6.5 Servidor Web	18
1.6.6 Sistema Gestor de Base de Datos	18
1.6.7 PgAdmin III 1.16.1	19
1.6.8 Herramientas CASE	19
1.6.9 Notación para el Modelado de Procesos de Negocio (Modelado Business Process Modeling Notation (BPMN))	20
1.6.10 Framework	20
1.6.11 Entorno de Desarrollo Integrado (IDE)	21
1.7 Conclusiones del capítulo	22

2	Propuesta de Solución	23
2.1	Introducción	23
2.2	Proceso del Vicedecanato de Investigación y Postgrado (VIP)	23
2.2.1	Subproceso 1: Conformar plan de indicadores real de la facultad	23
2.2.2	Subproceso 2: Actualizar indicadores y evidencia	24
2.2.3	Subproceso 3: Conformar Plan de Cursos	25
2.2.4	Subproceso 4: Solicitud de impartir un curso	26
2.2.5	Subproceso 5: Solicitud de matrícula a un curso	26
2.3	Características de la Propuesta del Sistema	27
2.3.1	Roles y Funcionalidades	28
2.4	Conclusiones del capítulo	29
3	Exploración, Planificación y Diseño del Sistema	30
3.1	Introducción	30
3.2	Exploración	30
3.2.1	Historias de Usuario	31
3.3	Planificación	41
3.3.1	Estimación de Esfuerzo	42
3.3.2	Plan de Iteraciones	43
3.3.3	Plan de Entrega	45
3.4	Diseño	45
3.4.1	Patrones de Arquitectura	45
3.4.2	Patrones de Diseño	47
3.4.3	Modelo de Datos	48
3.4.4	Tarjetas Clase Responsabilidad Colaboración (Clase Responsabilidad Colaboración (CRC))	49
3.5	Conclusiones del capítulo	53
4	Implementación y Pruebas	54
4.1	Introducción	54
4.2	Implementación	54
4.2.1	Código Fuente	55
4.2.2	Tareas de Ingeniería	56
4.3	Pruebas del Software	66
4.3.1	Pruebas Unitarias	66
4.3.2	Pruebas de Aceptación	67
4.4	Conclusiones del capítulo	72

Conclusiones	73
Recomendaciones	74
Acrónimos	75
Referencias bibliográficas	76
Bibliografía	79
Apéndices	84
A Pruebas Aceptación	85
A.1 Gestionar Anuncios	85
A.2 Gestionar Cursos	85
A.3 Insertar solicitud de impartir un curso	86
A.4 Gestionar Matrícula	87
A.5 Gestionar Solicitud de Evidencia	87
A.6 Gestionar Solicitud de Matrícula	88
A.7 Gestionar Solicitud de Curso	88
A.8 Generar Gráficas Estadísticas	89
A.9 Generar Informes del Estado de los Indicadores	89
A.10 Gestionar Descargas	90
A.11 Gestionar Planes de Área	90
A.12 Gestionar Solicitud de Anuncios	91
A.13 Consultar Ranking	91
A.14 Consultar Descargas	92
A.15 Consultar Cursos	92
A.16 Consultar Anuncios	93
A.17 Consultar Evidencias de Usuario	93
B Tarjetas CRC	95
B.1 LoginController	95
B.2 ExportarpdfController	95
B.3 DescargasController	96
B.4 DefaultController	96
B.5 GraficosController	96
B.6 AnuncioController	97
B.7 CursoController	97
B.8 InformeController	97

B.9 CategoriTrabajadorController 98

Índice de figuras

1.1	Interfaz web del SGI	10
1.2	Interfaz web de Investigación UTALCA	10
1.3	Interfaz web del SIGA	11
1.4	Interfaz web del Sistema de Indicadores Ciencia Tecnología e Innovación (SIndiCIT)	12
1.5	Interfaz web del Sistema para la Gestión de Evidencias de Indicadores de CTI y Postgrado (GEVIN)	12
1.6	Interfaz web del Sistema de gestión Universitaria	13
1.7	Interfaz web Serie Científica	14
2.1	Conformar plan de indicadores real de la facultad	24
2.2	Actualizar indicadores y evidencia	25
2.3	Conformar Plan de Cursos	25
2.4	Solicitud de impartir un curso	26
2.5	Solicitud de matrícula a un curso	27
3.1	Diagrama del Modelo Vista Controlador usado por Symfony.	47
3.2	Diagrama Entidad Relación	49
4.1	Resultados de la ejecución de los test unitarios	67

Índice de tablas

3.1	Historia de usuario # 1	31
3.2	Historia de usuario # 2	32
3.3	Historia de usuario # 3	32
3.4	Historia de usuario # 4	33
3.5	Historia de usuario # 5	33
3.6	Historia de usuario # 6	33
3.7	Historia de usuario # 7	33
3.8	Historia de usuario # 8	34
3.9	Historia de usuario # 9	34
3.10	Historia de usuario # 10	34
3.11	Historia de usuario # 11	35
3.12	Historia de usuario # 12	35
3.13	Historia de usuario # 13	36
3.14	Historia de usuario # 14	36
3.15	Historia de usuario # 15	36
3.16	Historia de usuario # 16	37
3.17	Historia de usuario # 17	37
3.18	Historia de usuario # 18	38
3.19	Historia de usuario # 19	38
3.20	Historia de usuario # 20	38
3.21	Historia de usuario # 21	39
3.22	Historia de usuario # 22	39
3.23	Historia de usuario # 23	40
3.24	Historia de usuario # 24	40
3.25	Historia de usuario # 25	40
3.26	Historia de usuario # 26	41
3.27	Historia de usuario # 27	41
3.28	Estimación de esfuerzo por historia de usuario	42
3.29	Plan de duración de las iteraciones	44
3.30	Plan de Entregas	45

3.31	Tarjeta CRC # 1	49
3.32	Tarjeta CRC # 2	49
3.33	Tarjeta CRC # 3	49
3.34	Tarjeta CRC # 4	50
3.35	Tarjeta CRC # 5	50
3.36	Tarjeta CRC # 6	50
3.37	Tarjeta CRC # 7	51
3.38	Tarjeta CRC # 8	51
3.39	Tarjeta CRC # 9	51
3.40	Tarjeta CRC # 10	51
4.1	Tarea de ingeniería # 1	57
4.2	Tarea de ingeniería # 2	57
4.3	Tarea de ingeniería # 3	58
4.4	Tarea de ingeniería # 4	58
4.5	Tarea de ingeniería # 5	58
4.6	Tarea de ingeniería # 6	58
4.7	Tarea de ingeniería # 7	59
4.8	Tarea de ingeniería # 8	59
4.9	Tarea de ingeniería # 9	59
4.10	Tarea de ingeniería # 10	60
4.11	Tarea de ingeniería # 11	60
4.12	Tarea de ingeniería # 12	61
4.13	Tarea de ingeniería # 13	61
4.14	Tarea de ingeniería # 14	61
4.15	Tarea de ingeniería # 15	61
4.16	Tarea de ingeniería # 16	62
4.17	Tarea de ingeniería # 17	62
4.18	Tarea de ingeniería # 18	63
4.19	Tarea de ingeniería # 19	63
4.20	Tarea de ingeniería # 20	63
4.21	Tarea de ingeniería # 21	63
4.22	Tarea de ingeniería # 22	64
4.23	Tarea de ingeniería # 23	64
4.24	Tarea de ingeniería # 24	64
4.25	Tarea de ingeniería # 25	65
4.26	Tarea de ingeniería # 26	65
4.27	Tarea de ingeniería # 27	65

4.28 Prueba de aceptación # 1	67
4.29 Prueba de aceptación # 2	68
4.30 Prueba de aceptación # 3	68
4.31 Prueba de aceptación # 4	69
4.32 Prueba de aceptación # 5	69
4.33 Prueba de aceptación # 6	70
4.34 Prueba de aceptación # 7	70
4.35 Prueba de aceptación # 8	71
4.36 Prueba de aceptación # 9	71
4.37 Prueba de aceptación # 10	72
A.1 Prueba de aceptación # 11	85
A.2 Prueba de aceptación # 12	85
A.3 Prueba de aceptación # 13	86
A.4 Prueba de aceptación # 14	87
A.5 Prueba de aceptación # 15	87
A.6 Prueba de aceptación # 16	88
A.7 Prueba de aceptación # 17	88
A.8 Prueba de aceptación # 18	89
A.9 Prueba de aceptación # 19	90
A.10 Prueba de aceptación # 20	90
A.11 Prueba de aceptación # 21	90
A.12 Prueba de aceptación # 22	91
A.13 Prueba de aceptación # 23	92
A.14 Prueba de aceptación # 24	92
A.15 Prueba de aceptación # 25	93
A.16 Prueba de aceptación # 26	93
A.17 Prueba de aceptación # 27	94
B.1 Tarjeta CRC # 11	95
B.2 Tarjeta CRC # 12	95
B.3 Tarjeta CRC # 13	96
B.4 Tarjeta CRC # 14	96
B.5 Tarjeta CRC # 15	96
B.6 Tarjeta CRC # 16	97
B.7 Tarjeta CRC # 17	97
B.8 Tarjeta CRC # 18	97
B.9 Tarjeta CRC # 19	98

Lista de códigos fuentes

4.1	Ejemplo de código en PHP	56
-----	------------------------------------	----

En el mundo civilizado de hoy la ciencia y la tecnología han llegado a ser irremplazables para disímiles sectores de la sociedad ¹. En estos sectores el uso y explotación de la información es prácticamente indispensable para el manejo de los sistemas y sus respuestas. La informatización de procesos manuales o semi-informatizados, es una de las tareas que más ha evolucionado el entorno a finales del siglo XX y lo que va del XXI. Ha provisto al mundo de nuevas herramientas informáticas para la manipulación, transferencia, control y almacenamiento de la información. Esto permite a la sociedad interactuar mediante sistemas cada vez más simplificados y que logran incluir en ellos un amplio espectro de posibilidades y nuevas funcionalidades.

El desarrollo de este enfoque en diferentes países ha ido permeando paulatinamente la esfera organizacional. Su adopción no constituye una tarea sencilla por cuanto las organizaciones requieren de una cultura informacional y de políticas expresas de reconocimiento del papel de la información y el conocimiento para el desarrollo organizacional (CHAÍN, 1999).

La gestión de información ha sido visualizada por muchas organizaciones como un enfoque que contribuye a la elevación de la eficacia y eficiencia organizacional, así como al perfeccionamiento de la toma de decisiones. Lo que ha tenido una presencia significativa en muchos países desarrollados, y en instituciones que han contado con líderes capaces de visualizar la importancia de contar con la información que necesitan en el momento oportuno. (ibíd.).

El Dr. C. Israel Núñez Paula de la Universidad de La Habana plantea que Probablemente la definición más citada sobre la gestión de información es la de (Linda Woodman, 1985): *“la gestión de la información es todo lo relacionado con: la obtención de la información adecuada, en la forma correcta, para la persona u organización indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta”* (NÚÑEZ y D. PÉREZ, 2013).

Obtener la información necesaria, con la calidad requerida, es una premisa indispensable para la supervivencia de las empresas, si se considera que las organizaciones acortan cada vez más sus ciclos estratégicos y que la toma de decisiones, así como el cambio, es continuo (AJA, 2002).

Como Cuba no está aislada de todo el acontecer tecnológico que sucede en el mundo, se ha dado a la tarea de potenciar su informatización. Para ello ha extendido por todo el país una serie de equipamientos que tecnológicamente ² han hecho posible a cada centro de trabajo e institución educacional explotar los

¹Dígase por estos sectores Educación, Salud, Telecomunicaciones, etc.

²Se refiere a la infraestructura tecnológica que se ha venido extendido a todas las provincias de Cuba.

beneficios de esta era digital. Este proceso va respaldado por el artículo 39 de la Constitución de la República, donde establece que *“el Estado estimula y viabiliza la investigación y prioriza la dirigida a resolver los problemas que atañen al interés de la sociedad y al beneficio del pueblo”* (CUBA, 1901).

Para llevar a cabo dicha informatización se ha delegado la tarea a instituciones desarrolladoras de software que han nacido en el país al calor de esta explosión tecnológica. Entre las más prestigiosas se encuentra la casa de altos estudios Universidad de las Ciencias Informáticas UCI, pilar universitario que sirve de soporte a la industria cubana de la informática. La universidad persigue formar todo un profesional de las ciencias informáticas. Para lograrlo integra dentro de su plan de estudio la docencia, producción e investigación. Como en todas las ramas laborales del país, le proporciona al personal graduado un seguimiento en su superación, ya que la mayor parte de su claustro son egresados de dicha institución.

La superación profesional y académica del claustro de profesores contribuye en gran medida a ubicar a cualquier universidad en el mundo en un nivel cada vez más prestigioso con respecto a la clasificación académica de universidades ³, aportando así un mayor impacto educativo, metodológico y profesional al estudiantado universitario.

La dirección de formación post-graduada es la dirección encargada de garantizar la superación profesional y académica del claustro de profesores de la UCI. La misma contribuye a la superación de los profesionales universitarios de todo el país en la rama de la Informática. Dicha dirección como principal objetivo tiene elevar la competitividad, desempeño y capacidad innovadora en la esfera de la investigación, la docencia y la producción de software de su claustro profesoral. Para ello a puesto en práctica mecanismos para sustentar dicho objetivo mediante la alianza con instituciones, empresas y organizaciones, fortaleciendo la imagen de la UCI en el ámbito nacional e internacional (UCI, 2015).

Los objetivos específicos del “Proceso de Investigación” en la UCI son (UCI, 2011):

1. Fortalecer la creación de unidades organizativas responsabilizadas con la innovación tecnológica dentro de las áreas de la universidad.
2. Lograr una mayor excelencia en los resultados científico-tecnológicos, expresada en el incremento de la pertinencia, la calidad y el impacto de los productos de exportación y nacionales que desarrolla la universidad.
3. Fortalecer la gestión de la educación permanente de los graduados universitarios, fundamentalmente en la rama de la informática, en cuanto al desarrollo de maestrías, especialidades y doctorados, sobre la base del aprendizaje y socialización del conocimiento, posibilitando una estrecha colaboración con instituciones nacionales e internacionales.
4. Asegurar la cooperación, nacional e internacional, con instituciones académicas y de investigación vinculadas a las materias a fines a cada centro.
5. Concebir el Sistema de Gestión de Calidad integrando las actividades principales como Desarrollo, Formación, Investigación y Postgrado.
6. Garantizar la realización de eventos científicos que se desarrollan en la UCI, organizados por la propia

³También se puede medir con respecto a otros aspectos.

universidad o por otra entidad.

Los objetivos específicos del “Proceso de Postgrado” en la UCI son (ibíd.):

1. Asegurar que se brinde a extranjeros, cursos de postgrado, maestría y doctorados y otros cursos especializados, cumpliendo con las normativas establecidas al respecto en el país.
2. Asegurar la colaboración académica y científica universitaria con otras universidades y entidades nacionales e internacionales en el marco de las regulaciones establecidas al respecto en el país.
3. Fortalecer la implementación de la estrategia para la superación profesional en la rama de la informática, promoviendo una amplia participación del claustro en los programas de superación académica; en particular la formación de doctores y máster.
4. Fortalecer las relaciones entre las direcciones de Postgrado y Ciencia y Técnica del Ministerio de Educación Superior (MES); así como con la red de Consejo de Educación Superior (CES) del país y otras instituciones científicas y académicas del país y extranjeras.
5. Establecer el Sistema de Gestión de la Calidad en las actividades de postgrado que posibilite la organización de los procesos y su mejora continua para satisfacer las expectativas de los profesores, contando con criterios de medidas para ejercer su control en función de las prioridades que se establezcan.
6. Garantizar un servicio de calidad a los procesos sustantivos de la universidad a través del empleo intensivo de la información científico-técnica utilizada.
7. Promover las sinergias entre los procesos de investigación científica, formación profesional, superación académica, información y producción que permita lograr resultados superiores.
8. Posibilitar la coordinación con la Dirección de Colaboración Internacional de políticas encaminadas a la obtención de proyectos de colaboración con universidades y entidades de otros países para financiar programas de superación académica y científica del claustro de profesores.
9. Fortalecer la gestión de las actividades del Ministerio de Ciencia, Tecnología y Medio Ambiente (CIT-MA).
10. Satisfacer con calidad las necesidades de formación de pregrado, postgrado y de capacitación en el área de las Ciencias Informáticas, de acuerdo con las prioridades del desarrollo del país.

La UCI actualmente cuenta con 7 facultades, dentro de las mismas está el área del VIP. Éste tiene la misión de implementar las estrategias, políticas, resoluciones y reglamentos de los sistemas de Ciencia y Técnica, y de la superación posgraduada, así como de la información científico técnica establecidas por la dirección de la universidad y los órganos rectores de la actividad, optimizando los recursos y mediante el uso adecuado de las tecnologías de información y de las comunicaciones (UCI, 2013). Parte de lo realizado para cumplir con lo anterior mencionado es gestionar la información referente a la Investigación y el Postgrado de sus profesores y especialistas, y almacenar sus evidencias relacionadas con Ciencia, Tecnología e Investigación (CTI).

Los procesos que maneja para dar seguimiento a ello son:

- Premios obtenidos.

- Publicaciones científicas.
- Patentes y registros.
- Proyectos I+D.
- Resultados introducidos.
- Trabajos presentados en eventos.
- Capacitación.
- Empleo de estudiantes.

Definido por la Dirección de Investigaciones de la universidad se utiliza actualmente el Sistema de Indicadores Ciencia Tecnología e Innovación SIndiCIT Figura 1.4, para gestionar los procesos que maneja el área. Al hacer un análisis de los procesos que SIndiCIT maneja, se detectan un grupo de necesidades específicas de la facultad que el mismo no incluye:

1. Permitir el acceso al sistema a todo el personal para consultar sus datos y realizar otras tareas según su rol: El sistema ha sido diseñado para que lo usen personas con determinados conocimientos y privilegios. Solo pueden acceder a él gestores que laboran en estas áreas. Para poder ingresar datos al mismo debe estar registrado en la aplicación y tener los permisos necesarios sobre la facultad que se quiere trabajar.
2. Gestionar las evidencias junto a los indicadores en la misma aplicación a través de archivos adjuntos: La gestión de las evidencias, se hace por separado a los indicadores y las mismas, son almacenadas en un servidor distinto al del SIndiCIT por acceso ftp (Protocolo de Transferencia de Archivos). Así divide algo que en la práctica es un solo proceso en dos flujos de información por separado.
3. Crear un espacio informativo y de consulta: No permite divulgar eventos científicos y tener el conocimiento de quién participa o no. No existe un módulo de publicación para profesores, donde se pueda ofertar cursos y a su vez que cada participante pueda matricularse a través del mismo. Esto ayudaría a confeccionar una plantilla de matrícula, y los cursos serían más abiertos al conocimiento de la comunidad.
4. Consultar información y evidencias personales y de mis subordinados según el área: El sistema no posibilita a los profesores consultar información personal, ya que no tienen acceso al mismo. Esto afecta a los jefes de departamentos y otras áreas que no tienen información sobre sus subordinados en el ámbito científico-investigativo.
5. Automatizar la actualización de los datos contables: No actualiza de manera automática todos los datos contables, es decir en la mayoría de los casos queda de parte del usuario realizar los cálculos e introducir los resultados manualmente. Cuando una evidencia o indicador es introducido, el encargado actualiza manualmente los datos incrementales que deban ser afectados con la nueva entrada.
6. Garantizar un mayor control y seguimiento a la superación del claustro desde los departamentos docentes y los centros de desarrollo de software.

Anteriormente, se han desarrollado trabajos de diploma sobre este tema, pero no ha quedado evidencia

práctica que solucione el problema actual. Solo en la facultad 1 existe un sistema llamado Sistema para la Gestión de Evidencias de Indicadores de CTI y Postgrado GEVIN Figura 1.5, que gestiona procesos parecidos, pero enfocado a las actividades de un centro productivo, en este caso el Centro de Ideoinformática (CIDI). Este sistema ya está implementado, aunque aún no está desplegado, mientras que en el resto de las facultades no existe ningún sistema.

Al hacer un análisis en esta área de la Facultad 2 se arrojó que el flujo de información tiene una estructura pesada, se trabaja de manera engorrosa ya que la información fluye a través de documentos o por vía correo. Esta forma de trabajar implica mucho más esfuerzo, tiempo y un aumento considerable en el margen de error pues puede tener como consecuencia la desorganización, desactualización, ineficiente almacenamiento y la pérdida de datos.

Teniendo en cuenta lo anterior expuesto se plantea el siguiente **problema a resolver**: ¿Cómo facilitar el proceso de Gestión de Información inherente al área del Vicedecanato de Investigación y Postgrado VIP en la Facultad 2?

Como **objeto de estudio** de esta investigación se tiene: Sistemas Informáticos para la Gestión de Información.

Campo de acción: Sistemas Informáticos para la Gestión de Información del proceso de Investigación y Postgrado.

Siendo el **objetivo general** de la presente investigación: Desarrollar un sistema informático para la gestión de información en el área del Vicedecanato de Investigación y Postgrado VIP de la Facultad 2.

Para dar cumplimiento al objetivo general se desglosaron los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación.
2. Definir las funcionalidades que debe cumplir el sistema.
3. Implementar un sistema informático para la gestión de información que cumpla con los requerimientos definidos.
4. Aplicar pruebas al sistema resultante para garantizar la calidad y fiabilidad del producto.

Para guiar el desarrollo de la presente investigación surgen las siguientes **preguntas científicas**:

- ¿Cuáles son las posiciones teóricas en cuanto a los sistemas para la gestión de información y su empleo en el área de Investigación y Postgrado?
- ¿Cuál es el estado real de desarrollo alcanzado en la informatización de las áreas involucradas en el flujo y manejo de información del VIP de la Facultad 2?
- ¿Cómo diseñar un sistema para la gestión de información que se ajuste a las características de los procesos que se manejan en el VIP de la Facultad 2?
- ¿Cómo implementar un sistema para la gestión de información que garantice agilidad y rapidez en la ejecución de los procesos que se manejan en el VIP de la Facultad 2?

Con el propósito de dar cumplimiento a lo anteriormente planteado, se elaboraron las siguientes **tareas de investigación**:

1. Revisión bibliográfica para generar el marco teórico conceptual en lo referente al desarrollo de los SGI.
2. Estudio de sistemas homólogos para conocer aspectos regulares en el diseño de los SGI aplicados al proceso de investigación y postgrado.
3. Análisis de las principales deficiencias y necesidades presentes en la forma actual de desarrollarse el proceso de gestión de información en el Vicedecanato de Investigación y Postgrado de la facultad 2, para diseñar el sistema.
4. Análisis de las herramientas informáticas y metodologías de desarrollo de software existentes para realizar la implementación del sistema.
5. Identificación de las principales funcionalidades del sistema para la posterior implementación del mismo, teniendo en cuenta los requisitos definidos anteriormente.
6. Investigación de los diferentes tipos de pruebas funcionales, para su posterior aplicación sobre el sistema desarrollado.

Métodos de Investigación Teóricos(COELLO, 2008)

- Permiten estudiar las características del objeto de investigación que no son observables directamente.
- Facilitan la construcción de modelos e hipótesis de investigación.
- Crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad.
- Posibilitan el conocimiento del estado del arte del fenómeno, su evolución en una etapa determinada y su relación con otros fenómenos.
- Se apoyan en el proceso de análisis y síntesis.

Analítico-Sintético(ibíd.)

Análisis: Permite la descomposición mental del objeto o fenómeno en sus múltiples relaciones o componentes para facilitar su estudio.

Síntesis: Establece mentalmente la unión entre las partes previamente analizadas, permite descubrir sus características generales y las relaciones esenciales entre ellas.

Aplicado para analizar la información obtenida en el proceso de investigación en el área del VIP de la facultad 2, posibilitando identificar aspectos que puedan ser empleados en el desarrollo de la presente investigación. Además para el análisis de la información existente dentro y fuera del país, acerca de los SGI. Identificando así, conceptos, definiciones, avances y otros elementos concluyentes de utilidad para la investigación en cuestión.

Inducción-Deducción(ibíd.)

Inducción: Es una forma de razonamiento a través del cual se pasa de un conocimiento de cosas particulares a un conocimiento más general que va a reflejar lo que hay de común en esos fenómenos individuales.

Deducción: Es un procedimiento que permite, a partir de conocimientos generales, inferir casos particulares por un razonamiento lógico.

Utilizado para el análisis de los comportamientos y algunas características de la gestión de información en el área del VIP. Y reunir varios elementos disponibles para poder deducir conclusiones sobre casos particulares que pueden ser verificados en la práctica.

Métodos Empíricos

Describen y explican las características fenomenológicas del objeto. Representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional(S. HERNÁNDEZ, 2008).

Entrevista

Es una conversación planificada para obtener información. Su uso constituye un medio para el conocimiento cualitativo de los fenómenos o sobre características personales del entrevistado y puede influir en determinados aspectos de la conducta humana por lo que es importante una buena comunicación (ibíd.).

Aplicado a los trabajadores del VIP para constatar las condiciones con que cuentan para la gestión de la información generada a partir de la actividad científica e investigativa de la facultad 2. Además aplicado a los jefes de los departamentos docentes, subdirectores de los centros de desarrollo, a profesores y especialistas del área para conocer cómo se gestiona y maneja la información referente a la investigación y al postgrado de la facultad 2.

Observación(ibíd.):

- Puede utilizarse en distintos momentos de la investigación.
- Permite la recogida de la información de cada uno de los conceptos.
- Permite investigar el fenómeno en su manifestación externa.
- El documento guía de la observación debe ser lo suficientemente preciso y claro para garantizar que diferentes observadores lo apliquen de igual forma.

Su puesta en práctica permite monitorizar de cerca el negocio, y analizar los resultados de la gestión de información en el área del VIP de la facultad 2.

Modelado

Es la reproducción simplificada de la realidad que permite descubrir nuevas relaciones y cualidades del objeto. Este método permite la creación de modelos, (propuestas, alternativas, estrategias, etc)(ibíd.).

En esta investigación a través del método empírico se realiza el modelado mediante diagramas. Esto permite reflejar la estructura, las relaciones y características de la solución propuesta. En este caso, con el uso de BPMN, se facilita también el diseño de clases necesario para la implementación de la aplicación.

1.1. Introducción

El mundo de las tecnologías mantiene hoy un constante crecimiento. Esto lleva al ser humano a mantenerse al día en las principales herramientas informáticas y procesos que se utilizan en el desarrollo de aplicaciones informáticas. Muchas veces, sin darse cuenta, se puede utilizar una tecnología o herramienta obsoleta, lo que resta a las aplicaciones en calidad, funcionamiento, prestigio, y confiabilidad. Teniendo en cuenta lo antes planteado el presente capítulo está destinado a brindar una breve descripción de todo lo relacionado con los conceptos asociados a la gestión de información. Se expondrá además un estudio del estado del arte, recorriendo el ámbito internacional y nacional de sistemas informáticos que realizan acciones similares de gestión de información referente a la investigación y el postgrado. Además se realiza un estudio de las principales tecnologías en las que se apoya el desarrollo de estos tipos de sistemas en función de un análisis de las tendencias actuales.

1.2. Conceptos Asociados

1.2.1. Gestión de Información

Probablemente la definición más citada sobre la Gestión de Información es la de (Linda Woodman, 1985), en la cual plantea que *“La gestión de la información es todo lo relacionado con: la obtención de la información adecuada, en la forma correcta, para la persona u organización indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta”* (NÚÑEZ y D. PÉREZ, 2013).

La finalidad de la Gestión de la información es ofrecer mecanismos que permitieran a la organización adquirir, producir y transmitir, al menor coste posible, datos e informaciones con una calidad, exactitud y actualidad suficientes para servir a los objetivos de la organización. En términos perfectamente claros la Gestión de Información se entiende por tener la información adecuada, para la persona que lo necesita, en el momento que lo necesita, al mejor precio posible para toma la mejor de las decisiones. (AJA, 2002).

1.2.2. Sistema de Gestión de Información SGI

Un sistema de información es un conjunto de componentes que interactúan entre sí para satisfacer las necesidades de información en un entorno determinado, dígame una organización. Estos componentes pueden ser personas, datos, actividades o recursos materiales en general. Los cuales procesan la información y la distribuyen de manera adecuada, buscando satisfacer las necesidades de la organización. El objetivo primordial de un sistema de información es apoyar la toma de decisiones y controlar todo lo que en ello ocurre.

Con lo antes expuesto se concluye que, un SGI permite la gestión de los recursos de información tanto internos como externos. Su finalidad es generar servicios y productos que respondan a las necesidades y sobrepasen las expectativas de los usuarios, posibilitando que el sistema trabaje eficientemente y económicamente a la vez. El SGI aprovecha al máximo sus recursos de información en función de la mejora continua y de la toma de decisiones de la organización a todos los niveles jerárquicos desde la cúspide estratégica hasta la base operativa (V. FERNÁNDEZ, 2008).

Sistema Informático

El concepto más completo y preciso de Sistema Informático según la Organización Internacional de Normalización (ISO) es: Conjunto de elementos que hacen posible el tratamiento automático de la información.

1.2.3. Sistema Informático de Gestión de Información

SGI: “Sistema integrado y automatizado para proveer la información que sostenga las funciones de operatividad, gestión y toma de decisiones en una organización”(HAMPTON, 1985).

Luego de conocer los conceptos de sistema informático, sistema informático de gestión de información, sistema de gestión de información, gestión de información podemos definir como Sistema Informático de Gestión de Información, es aquel sistema integrado e informatizado que provee los datos que sostienen las funciones de operatividad, gestión y toma de decisiones en una organización.

1.3. Estudio del Estado del Arte

1.3.1. Internacional

Sistema de Gestión de la Investigación en la Universidad de Talca, Chile (SGI)

El sistema ver en la Figura 1.1 es capaz de mantener actualizadas las publicaciones de programas y proyectos de investigación, proyectos de tesis y sus consiguientes resultados, sean estos, publicaciones, presentaciones en congresos disciplinarios, organización de eventos, u otros. Cuenta con un directorio investigador donde se encuentran los principales datos de su personal investigativo (ALLENDES, 2015).

SGI DIRECCIÓN DE INVESTIGACIÓN VICERRECTORÍA ACADÉMICA INVERSIÓN EN LA INVESTIGACIÓN

WEB DI | DIRECTORIO INVESTIGADOR

..: Búsqueda de Investigadores

Nombre Investigador	Facultad/Vicerrectoría/Instituto	Unidad	Información
ACEVEDO OPAZO CESAR ANTONIO	Facultad De Ciencias Agrarias	Departamento De Produccion Agricola	(i)
ACUÑA SAN MARTIN MARCELA YANET	Facultad De Ciencias Jurídicas Y Sociales	Departamento De Derecho Privado	(i)
AEDO ORTIZ DARIO MARCELO	Facultad De Ciencias Forestales	Departamento De Produccion Forestal	(i)
AGUIRRE GONZALEZ MEDARDO JUAN	Facultad De Ciencias Empresariales	Escuela De Ingenieria Comercial	(i)
AGUIRRE LEON EDUARDO ALBERTO	Escuela De Arquitectura	Escuela De Arquitectura	(i)
ALARCON LEIVA JORGE ALBERTO	Instituto De Investigacion Y Desarrollo Educacional	Instituto De Investigacion Y Desarrollo Educacional	(i)
ALARCÓN LOZANO MARCELO ALEJANDRO	Facultad De Ciencias De La Salud	Departamento De Bioquímica Clínica E Inmunohematología	(i)
ALBORNOZ VERDUGO MARIA ELIANA	Facultad De Ciencias De La Salud	Departamento De Ciencias Basicas Biomedicas Y Microbiología	(i)
ALLENDES OSORIO RODOLFO	Facultad De Ingeniería	Departamento De Cs.De La Computacion	(i)

Universidad de Talca.

Figura 1.1. Interfaz web del SGI

Investigación UTALCA

La Universidad de Talca utiliza esta web ver en la Figura 1.2 para evidenciar y divulgar las investigaciones que realizan sus académicos por la vía de un sitio que sirva de punto de encuentro entre la oferta investigativa de la Universidad y la demanda de investigación de la sociedad y las empresas (FUENTES, 2014).

INVESTIGACION EN LA UTALCA

UNIVERSIDAD DE TALCA

INVESTIGACIÓN | PLATAFORMAS | RESULTADOS | RECURSOS

HOME | MAPA DEL SITIO | CONTACTO

DIRECCIÓN DE INVESTIGACIÓN

BÚSQUEDA DE ABSTRACT

PROGRAMA PIEI-ES

CENTROS TECNOLÓGICOS

INSTITUTOS

CENTROS DE ESTUDIOS

REVISTAS

RESULTADOS FONDECYT

Utalca potenciará el postgrado y la investigación
16 enero 2015
Al inaugurar las x jornadas de investigación, la rectora subrogante, Gilda Carrasco, dio a conocer, entre otros ajustes, la creación de la unidad de ciencia y tecnología y del centro

En Chile hay oportunidades para el desarrollo de la ciencia y tecnología
4 noviembre 2014
La autoridad reconoció que existe una alta concentración de

Figura 1.2. Interfaz web de Investigación UTALCA

Sistema Integral de Gestión Académica (SIGA)

Aplicación informática ver en la Figura 1.3 usada por la Universidad de Córdoba para la gestión académica. Abarca desde la gestión del acceso a la universidad hasta la tramitación de los títulos, la matrícula, actas de examen, expedientes, estadísticas, informes, gestión de becas y convalidaciones (F. CASTILLO, 2015).

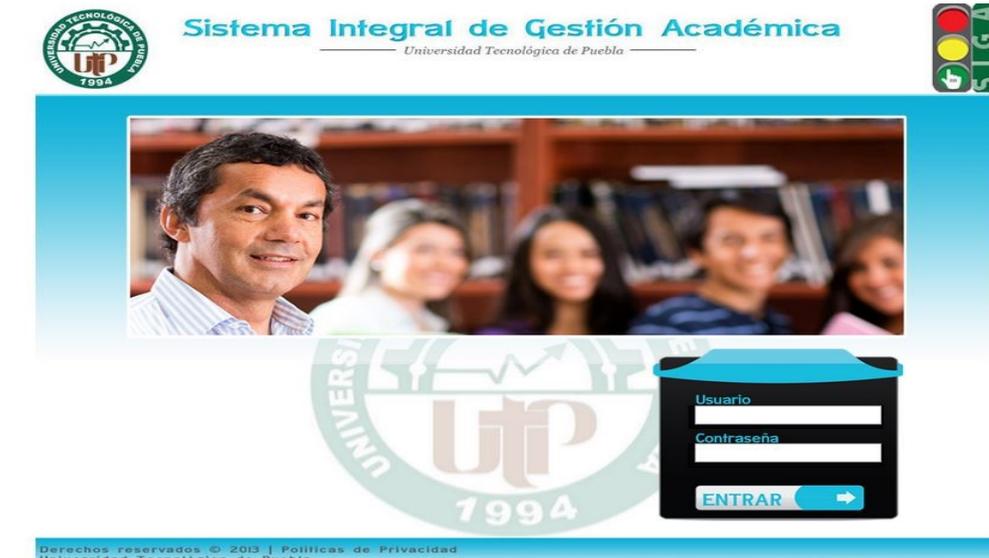


Figura 1.3. Interfaz web del SIGA

1.3.2. (UCI)

Sistema de Indicadores Ciencia Tecnología e Innovación SIndiCIT

Este sistema ver en la Figura 1.4, para la gestión de información, tiene implementado las funcionalidades que permiten:

1. Controlar solo de forma numérica el cumplimiento de los indicadores de CTI y datos de la actividad de postgrado de cada una de las áreas de la universidad.
2. Calcular una serie de parámetros preestablecidos, los cuales pueden dar una idea de cómo se encuentra la institución en ciencia e investigación.

Sistema para la Gestión de Evidencias de Indicadores de CTI y Postgrado (GEVIN)

Aunque es el sistema ver en la Figura 1.5 que más se acerca a la solución que plantea esta investigación no resuelve el problema planteado, ya que está implementado para su uso en un centro productivo específico CIDI. No realiza de manera automática el cierre de los cursos de postgrado que se están ofertando. No gestiona el trabajo con la información referente a eventos, publicaciones, cursos, maestrías, doctorados, ca-



Figura 1.4. Interfaz web del SINDICIT

pacitaciones, premios, trabajos presentados, patentes y registros, y empleo de estudiantes, sino que solamente se limita a guardar las evidencias como forma de almacenar las mismas y realizar posteriores consultas.

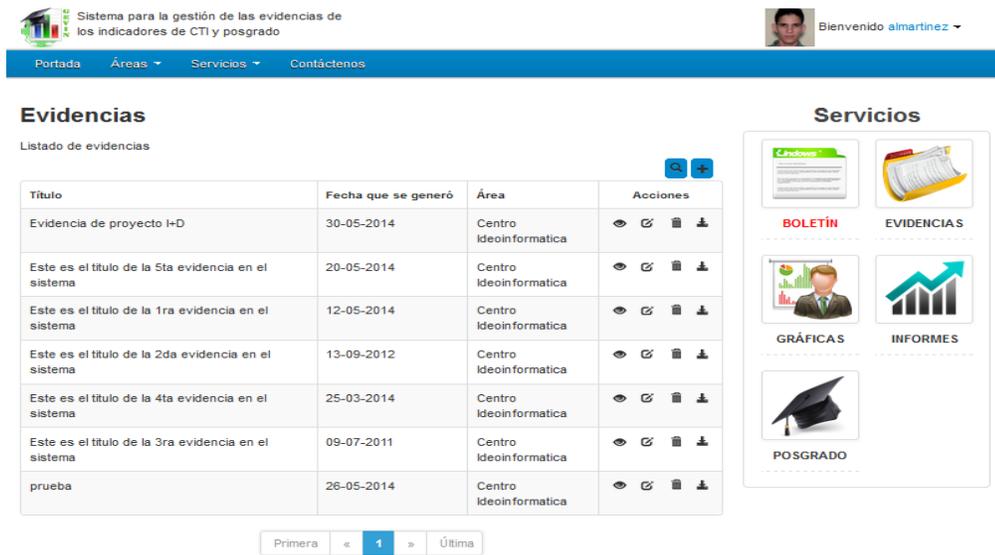


Figura 1.5. Interfaz web del GEVIN

Sistema de gestión Universitaria

Desarrollado por la dirección de Informatización. Perteneciente a la Vicerrectoría de Tecnología de la UCI ver en la Figura 1.6. Gestiona un grupo de información de las áreas docentes, productivas e investigati-

vas de la universidad referente a Pregrado, Residencia, Postgrado, Cooperación, Teleformación, Biblioteca, Desarrollo, Tecnologías, Investigación, Extensión, Ingreso y Egreso (UCI, 2014).



Figura 1.6. Interfaz web del Sistema de gestión Universitaria

Serie Científica

La Serie Científica de la Universidad de las Ciencias Informáticas (SC-UCI) ver en la Figura 1.7 es una publicación periódica académica diseñada para divulgar y promover los resultados de las investigaciones acerca de las diversas temáticas desarrolladas en la universidad. Se publica trimestralmente de forma electrónica y todos sus procesos editoriales se gestionan sobre la plataforma Open Journal Systems. Sus contenidos están disponibles libremente mediante autenticación desde la intranet universitaria, facultades regionales y centros de desarrollo.

Anualmente se publican de forma impresa los “Resúmenes de la Serie Científica de la Universidad de las Ciencias Informáticas” y la “Selección de artículos de la Serie Científica de la Universidad de las Ciencias Informáticas” que se distribuyen gratuitamente a bibliotecas universitarias cubanas y a centros e instituciones de investigación relacionados con las Ciencias de la Computación. Ambos títulos se encuentran registrados ante la Cámara Cubana del Libro y número de ISBN asignado.

Aunque está dirigida a los investigadores y profesionales de la Universidad de las Ciencias Informáticas acepta trabajos de autores adscritos a otras organizaciones siempre que estos estén en concordancia con las líneas de investigación (GARCÍA, 2014).



Figura 1.7. Interfaz web Serie Científica

1.4. Conclusiones del estudio

A partir del estudio realizado a los sistemas anteriores que de una forma u otra se relacionan con actividades de Investigación y Postgrado, se determina no utilizar ninguno de ellos para apoyar la investigación, debido a que no cubren las necesidades específicas detectadas en el VIP de la facultad 2. Los mismos no es que trabajan de manera ineficiente, ni tampoco que no realizan las acciones para lo que fueron creados pero no responden a las necesidades específicas del área. Por tanto no dan solución al problema planteado en esta investigación. A raíz de lo estipulado anteriormente se dispone a implementar un SGI que cubra las necesidades detectadas en el VIP de la facultad 2, y capaz de satisfacer los requerimientos del negocio exigidos por el cliente.

1.5. Metodologías de Desarrollo de Software

1.5.1. Programación Extrema XP

El desarrollo de software no es una tarea fácil. Prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte, se tienen aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas informáticas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos, pero también han presentado problemas en otros muchos. Una posible mejora es incluir en los procesos de desarrollo más actividades, más artefactos y más restricciones, basándose en los puntos débiles detectados. Sin embargo, el resultado final sería un proceso de desarrollo más complejo que puede incluso

limitar la propia habilidad del equipo para llevar a cabo el proyecto.

Otra aproximación es centrarse en otras dimensiones, como por ejemplo el factor humano o el producto software. Esta es la filosofía de las metodologías de desarrollo de software ágiles, las cuales dan mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas. Este enfoque está mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo pero manteniendo una alta calidad. Las metodologías de desarrollo de software ágiles están revolucionando la manera de producir software y, a la vez, generando un amplio debate entre sus seguidores y quienes, por escepticismo o convencimiento, no las ven como una alternativa para las metodologías tradicionales (LETELIER y PENADÉS, 2006).

La presente investigación es un proyecto pequeño, de poca duración, se tiene un equipo de desarrollo pequeño, y el cliente es parte de la investigación. Por tales motivos el equipo de desarrollo se inclina al uso de una metodología de desarrollo de software ágil. Dentro de este tipo de metodología de desarrollo de software se escoge utilizar XP:

- Es el más destacado de los procesos ágiles de desarrollo de software.
- Se considera como la adopción de las mejores metodologías de desarrollo.
- Los cambios de requisitos sobre la marcha son un aspecto natural.
- Plantea la programación en parejas.
- Apoya la integración del equipo de programación con el cliente o usuario.
- Consta de una alta gama de bibliografía. Y sus artefactos presentan una descripción del comportamiento del sistema fácil de entender.

Al estudiar las bibliografías de Kent Beck, padre de la metodología de desarrollo de software XP, se pueden definir seis fases para el uso de la misma: exploración, planificación, diseño, implementación y pruebas, mantenimiento, y muerte del proyecto. Por el alcance de esta investigación, la cantidad de integrantes del equipo y atendiendo a la necesidad respecto al tiempo en que se debería construir el sistema esta investigación solo abarca las cuatro primeras fases.

1.6. Herramientas y lenguajes informáticos

1.6.1. Lenguaje de Marcado de Hipertexto(HTML) 5

HTML es un lenguaje informático de composición de documentos que cuenta con instrucciones especiales que son interpretadas por el navegador y como resultado se muestran textos, imágenes, vídeos y enlaces. Utilizando HTML se puede definir la apariencia de las páginas creadas de diferentes formas. Contiene una gran variedad de etiquetas que son las encargadas de mantener la estructura e indicar la semántica de un documento, convirtiéndose en un lenguaje informático para presentación con soporte de aspectos visuales (MUSCIANO y KEMEDY, 1999).

Es un estándar reconocido en todo el mundo, multiplataforma, soportado por muchos navegadores. Con él, se puede obtener un producto atractivo y rápido sin mucho esfuerzo por parte del desarrollador. Permite establecer enlaces entre diferentes documentos y la introducción de referencias a otras páginas por medio de enlaces de hipertexto. HTML5 es la actualización de HTML, el lenguaje informático en el que es creada la web. HTML5 también es un término de marketing para agrupar las nuevas tecnologías de desarrollo de aplicaciones web: HTML5, CSS3 y nuevas capacidades de Javascript. La versión anterior de HTML, HTML4, carece de características necesarias para la creación de aplicaciones modernas basadas en un navegador y a su vez, el uso fuerte de Javascript ha ayudado a mejorar esto (J. VEGA, 2011).

Se decide utilizar HTML5 ya que éste presenta mejoras con respecto a las anteriores versiones de HTML. Además incluye un nuevo conjunto de etiquetas para dar estructura a un documento HTML, abandonando el uso de tablas y etiquetas div para la maquetación de las páginas.

En esta versión, los formularios son mejorados con una redefinición de nuevos tipos de elementos para la etiqueta input, y han sido incorporados al lenguaje nuevos atributos para los elementos de formulario (HOGAN, 2011).

1.6.2. Hojas de estilo en cascada(CSS) 3

Hojas de estilo en cascada (CSS por sus siglas en inglés) es un lenguaje informático de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML que separa los contenidos y su presentación. En la actualidad se utiliza el lenguaje informático CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, así como la posición de cada elemento dentro de la página.

Ventajas (EGUILUZ, 2009):

- Flexibilidad.
- Compatibilidad y continuidad.
- Separación del contenido y presentación.
- Unificación del diseño de las páginas del sitio.
- Optimización de los tiempos de carga y el tráfico del servidor.

Estas hacen al entorno de la aplicación más atractivo y amigable para los usuarios y permite a los diseñadores reducir los costos de mantenimiento de las páginas web al reducir la complejidad de las mismas.

Por lo anterior descrito se escoge CSS 3 para la aplicación de estilos a los contenidos que se muestran en el sistema a desarrollar.

1.6.3. JavaScript

Desarrollado por Netscape,Inc, JavaScript no forma parte de la plataforma Java; comparte algunos atributos con el lenguaje de programación Java, pero se desarrolló de forma independiente. Es un lenguaje

informático interpretado que permite incluir macros en páginas Web. Estas macros se ejecutan en el ordenador del visitante de nuestras páginas, y no en el servidor. Aspecto interesante porque los servidores Web suelen estar sobrecargados, mientras que los PC's de los usuarios no.

Se escoge JavaScript para el desarrollo de la presente solución informática porque proporciona los medios para:

- Controlar las ventanas del navegador y el contenido que muestran.
- Programar páginas dinámicas simples.
- Evitar depender del servidor Web para cálculos sencillos.
- Capturar los eventos generados por el usuario y responder a ellos sin salir a Internet.
- Optimización de los tiempos de carga y el tráfico del servidor.
- Comprobar los datos que el usuario introduce en un formulario antes de enviarlos.
- Comunicarse con el usuario mediante diversos métodos.

La característica de JavaScript que más simplifica la programación es que, aunque el lenguaje soporta cuatro tipos de datos, no es necesario declarar el tipo de las variables, argumentos de funciones ni valores de retorno de las funciones. El tipo de las variables cambia implícitamente cuando es necesario, lo que dificulta el desarrollo de programas complejos, pero ayuda a programar con rapidez macros sencillas.

El lenguaje JavaScript se inserta en documentos HTML, de forma que su código queda reflejado en la propia página y no es llamado o cargado de ninguna fuente externa. Se trata de un lenguaje interpretado puro y sensible a mayúsculas, aunque algunas implementaciones ignoran en parte este último extremo (GUERRERO y JUDITH, 2010).

1.6.4. Hypertext Pre-Processor (PHP) 5

PHP es un lenguaje informático muy potente, mundialmente utilizado en la programación de aplicaciones web del lado del servidor. Es similar a otros lenguajes informáticos como C o C++, rápido, de sintaxis cómoda y su sencillez contribuye a su rápido aprendizaje. Es un lenguaje interpretado, multiplataforma, se le pueden agregar extensiones fácilmente y dispone de una gran cantidad de librerías. Se puede utilizar como módulo de Apache, lo que lo hace extremadamente veloz. Por estar completamente escrito en C, se ejecuta rápidamente utilizando poca memoria. No hay que pagar licencias, no limita su distribución y se puede ampliar con nuevas funcionalidades si se desea. PHP 5 es la versión más adelantada, completa e intuitiva. El código es fácil de entender y mantener. Posee una vasta gama de funciones que le permiten adaptarse a cualquier entorno y a cualquier sistema operativo, por lo que es muy eficiente.

El objetivo principal de PHP 5 es perfeccionar el lenguaje, utilizando los mecanismos de programación orientada a objetos (POO) que no estaban presentes en las versiones anteriores, parámetro necesario e imprescindible que convierte a PHP en un lenguaje idóneo para todo tipo de aplicaciones y entornos. Es un lenguaje informático muy asequible, tanto para aquellas personas que tienen experiencia en la programación de sistemas web dinámicos, como para los que no.

PHP unido al servidor web Apache, representan las dos herramientas informáticas claves para la creación de sistemas o páginas web. Su excelente forma de trabajar con casi todas las bases de datos, hace de PHP el lenguaje informático idóneo para implementar un sistema de gestión de información. (A. CASTILLO, 2006). Se decide utilizar PHP en su versión 5.3.8 como lenguaje de programación, por lo anterior reflejado, y por la selección de Symfony como framework de desarrollo. Este último está basado en PHP 5.

1.6.5. Servidor Web

Apache 2.2.21

Fue escogido el servidor web Apache en su versión 2.2.21, debido a su configurabilidad, robustez y estabilidad, lo que hace de éste un servidor muy confiable. La licencia Apache es una descendiente de las licencias BSD (Distribución de Software Berkeley). La selección de esta tecnología como servidor web, está respaldada por las siguientes características que posee (G. GONZÁLEZ, 2014):

- Es posible su ejecución en una multitud de sistemas operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita de código abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Ésto le da una transparencia a este software de manera que si se quiere ver qué es lo que se instala como servidor, se pueda saber, sin ningún secreto, sin ninguna puerta trasera.
- Es un servidor altamente configurable de diseño modular. Es muy sencillo ampliar las capacidades del servidor web Apache. Actualmente existen muchos módulos para Apache que son adaptables a éste, y están ahí para que sean instalados cuando sea necesario.
- Trabaja con gran cantidad de lenguajes informático como Perl, PHP y otros lenguajes de script.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto.
- Posee una alta configurabilidad en la creación y gestión de registros. Permite la creación de ficheros de registro a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en el servidor.

1.6.6. Sistema Gestor de Base de Datos

Un Sistema Gestor de Base de Datos (SGBD), consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. De igual forma es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma (H. FERNÁNDEZ, 2009).

PostgreSQL 9.2

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD (Distribución de Software Berkeley) y con código fuente disponible libremente. Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multi-hilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

Se seleccionó PostgreSQL en su versión 9.2 teniendo en cuenta la necesidad de utilizar herramientas informáticas libres para el desarrollo de software. Además de que es un gestor multiplataforma, confiable, estable, con gran escalabilidad, control de concurrencia y funcionalidades que lo destacan como uno de los SGBD más potentes en la actualidad (OCAMPO y MONTROYA, 2013).

1.6.7. PgAdmin III 1.16.1

Ha sido seleccionado el PgAdmin III en su versión 1.16.1 dado que es una aplicación de diseño y manejo de bases de datos para su uso con PostgreSQL. Está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados (ALIAGA, 2008).

1.6.8. Herramientas CASE

Las herramientas CASE (Computer Aided Software Engineering) nacen para auxiliar a los desarrolladores de sistemas, lo que permite apoyo computarizado en todo o en parte del ciclo de vida de desarrollo de un sistema de software. Este tipo de herramientas informáticas facilitan el proceso de desarrollo de software con el aumento de la productividad y la reducción del tiempo que se utiliza. Utilizándolas se puede diseñar, implementar a partir del diseño realizado y brindarle seguridad al equipo de trabajo sobre el avance de la solución (DÍAZ, 2009).

Visual Paradigm 8

Es una herramienta informática profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Ayuda a una construcción más rápida de aplicaciones con mayor calidad y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso y generar código desde sus diagramas. (ibíd.).

Visual Paradigm para BPMN posibilita un ahorro considerable de tiempo y una calidad óptima en el modelado de procesos del negocio. Se considera muy completa y fácil de usar, con soporte multiplataforma y excelentes facilidades de interoperabilidad con otras aplicaciones.

Se decide utilizar esta herramienta informática para el modelado de procesos del negocio por las ventajas que ofrece (SIERRA, 2006):

- Generación de código: Modelo a código, diagrama a código, para diferentes lenguajes informáticos, entre ellos C++, Java y exportación como HTML.
- Generación de bases de datos: Permite la generación automática de bases de datos a partir de un Modelo entidad-relación.
- Interoperabilidad entre diagramas: Permite a partir de un diagrama obtener otro que guarde relación con el mismo.
- Tiene apoyo adicional en cuanto a generación de artefactos automáticamente.
- Disponibilidad en múltiples plataformas: Microsoft Windows (98, 2000, XP, o Vista), Linux, Mac OS X, Solaris o Java.

1.6.9. Notación para el Modelado de Procesos de Negocio (BPMN)

Business Process Modeling Notation o en español Notación para el Modelado de Procesos de Negocio (BPMN) es un estándar mundialmente reconocido, aceptado y aplicado en el mercado, para la diagramación y especificación de procesos de negocio, desde la modelización de procesos conceptuales y lógicos, hasta diseños de procesos orientados a tecnología Workflow. BPMN es una notación gráfica que describe la lógica de los pasos de un proceso de negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades (VENTURA y ESTEFANÍA, 2013).

El objetivo primario del lenguaje estándar BPMN es proveer una notación que sea legible y entendible para todos los usuarios de negocios, desde los analistas que realizan el diseño inicial de los procesos y los responsables de desarrollar la tecnología que ejecutará estos procesos, hasta los gerentes de negocios encargados de administrar y realizar el monitoreo de los procesos. BPMN define un modelo de procesos de negocio basándose en diagramas de flujo. Un modelo de procesos de negocio, es una red de objetos gráficos que representan las actividades y los controles de flujo que definen su orden de ejecución (WHITE, 2004).

Los elementos utilizados para construir los modelos BPMN fueron elegidos para ser distinguibles unos de otros y utilizar las figuras que son familiares a la mayoría de los diseñadores. Por ejemplo, las actividades se representan mediante rectángulos y las decisiones mediante rombos (VENTURA y ESTEFANÍA, 2013). Por lo anterior expuesto se decide escoger Notación para el Modelado de Procesos de Negocio BPMN para esta investigación.

1.6.10. Framework

En el desarrollo de software, un framework o marco de trabajo, es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje informático interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto (SÁNCHEZ, 2006).

La utilización de un framework, simplifica el desarrollo de un sistema mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Evita la re-implementación de funcionalidades que resultan ser frecuentes en este tipo de sistemas y que tradicionalmente incluyen administración de usuarios, persistencia de datos, motores de plantillas, entre otras. Facilita la programación de aplicaciones, puesto que encapsula operaciones complejas en instrucciones sencillas. Y además, proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener (DE LOS ANGELES, 2012).

Symfony 2.5

Es un completo framework diseñado para optimizar el desarrollo de las aplicaciones web mediante algunas de sus principales características. Para empezar, separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas informáticas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. El resultado de todas estas ventajas es que desde la propia instalación del framework, ya se tiene una estructura definida, así como la implementación de muchas funcionalidades, por lo que es innecesario cada vez que se crea una nueva aplicación web, preocuparse por la estructura que deba tener esta.

Symfony está desarrollado completamente con PHP 5. Ha sido probado en numerosos proyectos reales y se utiliza en sitios web de comercio electrónico de primer nivel. Es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Se puede ejecutar tanto en plataformas basadas en Unix como en plataformas Windows. Es fácil de instalar y configurar en la mayoría de plataformas. Además es fácil de extender, lo que permite su integración con las librerías de otros fabricantes. (LLONCH, 2013)

Se elige Symfony en su versión 2.5.0 como framework de desarrollo para la implementación de la solución propuesta, porque reúne las mejores prácticas de desarrollo web e integra muchas herramientas informáticas que aportan fortaleza al producto final. Usa además Doctrine o Propel como ORM (Object Relational Mapping) a opción del desarrollador. Unido a lo plasmado anteriormente las herramientas de generación de código, la interfaz de línea de comando para la instalación del sistema desarrollado, la amplia documentación que posee en Internet y otras tareas comunes automatizadas convierten a Symfony en un potente framework para PHP.

1.6.11. Entorno de Desarrollo Integrado (IDE)

PhpStorm 8

PhpStorm es diseñado específicamente para facilitar el desarrollo de aplicaciones web escritas en PHP. Sus principales características se pueden dividir en Navegación, Edición y re-factorización, archivos de sincronización y Depuración.

Se escoge PhpStorm como Entorno de Desarrollo Integrado ya que la aplicaciones web a desarrollarse consta de muchos archivos, bibliotecas, módulos y clases almacenadas en carpetas anidadas lo que se hace fácil con el uso del mismo. Además con PhpStorm los archivos abiertos se presentan en pestañas. Se mantiene un control en forma de árbol con carpetas y archivos. La Marco Estructura Modelo Vista Controlador (MVC) presenta el proyecto organizado en los controladores y vistas. El editor incorporado de PhpStorm es sensible al contexto y cuenta con plantillas personalizables, código de autocompletado y formato automático y plegable. Lo que ofrece el control total del código introducido (GAJDA, 2013).

1.7. Conclusiones del capítulo

Llevar a cabo el desarrollo de una aplicación necesita antes un exhaustivo trabajo de mesa para que los resultados sean los esperados. En este capítulo se analizan a profundidad los principales conceptos teóricos relacionados con la investigación. Luego de realizar un estudio del estado del arte de los SGI a nivel mundial, en nuestro país, y hasta en la propia universidad, se llega a la conclusión de que ninguno cubría todas las necesidades que requiere el problema planteado en esta investigación.

Se proponen las herramientas informáticas y metodologías de desarrollo de software que se utilizarán en la elaboración del sistema informático. Y tomando en cuenta que el país está inmerso en el proceso de migración a software libre, las herramientas informáticas seleccionadas, en su mayoría, se identifican por ser libres (de código abierto) y multiplataforma.

Por lo antes expuesto y a partir de las necesidades y dificultades existentes en la realización de los procesos relacionados con las actividades del VIP de la Facultad 2, se detalla en este capítulo todos los elementos teóricos que sustentan la solución del problema:

- Tomar como metodología de desarrollo de software ágil XP.
- Para la programación por parte del cliente utilizar Java Script y HTML 5 con el uso de CSS 3 y por parte del servidor PHP 5.
- Apache 2.2.21 como servidor Web.
- PostgreSQL como sistema gestor de base de datos en su versión 9.2.
- PG Admin III 1.16.1 como entorno de escritorio visual para PostgreSQL.
- Para el Modelado, BPMN, y como herramienta case el Visual Paradigm en su versión 8.0.
- Para el desarrollo de la aplicación el framework Symfony en su versión 2.5.
- Como entorno de Desarrollo Integrado (IDE) PhpStorm 8.
- Modalidad o arquitectura Cliente-Servidor y el patrón de arquitectura Modelo Vista Controlador.

2.1. Introducción

La creación de un SGI es una tarea difícil para cualquier equipo de desarrollo. Las soluciones propuestas deben responder a las necesidades del cliente. En éste proceso se identifican varias etapas, que van desde la declaración del problema y los requisitos del sistema, hasta la propuesta de solución del mismo. En éste capítulo se define la propuesta del sistema a desarrollar, con el propósito de satisfacer el objetivo de la investigación. Se describe el flujo actual del proceso en el VIP de la Facultad 2. Se hace referencia a los procesos de negocio que se desean informatizar y se especifican los aspectos funcionales del software que se propone construir.

2.2. Proceso del Vicedecanato de Investigación y Postgrado (VIP)

La modelación de proceso de negocio permite realizar una exploración del dominio del problema, con el fin de lograr comprensión por parte del equipo de desarrollo de los procesos que se realizan actualmente en la entidad y la relación que existe entre estos. Durante este proceso se van determinando necesidades operacionales, así como restricciones que presenta la entidad, obteniéndose finalmente un entendimiento del negocio para dar paso a la fase inicial del sistema. Además permite comprender las características del negocio a través de la descripción de los procesos del negocio (ESPINOSA, 2009).

El proceso que se maneja en el VIP es: la gestión de información de los indicadores de CTI y Postgrado en la facultad 2. En ésta área de la facultad el Vicedecano de Investigación y Postgrado (VIP1) es el responsable de gestionar esa información. A continuación se desglosa en 5 subprocesos, dicho proceso.

2.2.1. Subproceso 1: Conformar plan de indicadores real de la facultad

Inicialmente se tiene un plan de indicadores CTI y postgrado que debe cumplir la facultad, el cual contiene las actividades de carácter científica a desarrollarse en un período determinado de tiempo. El VIP1

se encarga de conformar un plan para cada área ¹ de la facultad en correspondencia al cumplimiento de la misma. Le envía el plan a los Jefes de Área ² y éstos revisan el posible cumplimiento de su área según su claustro. Le informan a cada trabajador su plan de indicadores y ellos revisan el posible cumplimiento de su plan individual. Si en alguno de estos casos no es posible cumplir el plan, se regresa en ese mismo orden, de Trabajador a Jefe de Área o de Jefe de Área a Vicedecano para una revisión y modificación del plan. El proceso termina cuando el trabajador acepta el cumplimiento de su plan o se rechaza el inconveniente planteado por el mismo. La Figura 2.1 muestra el modelado de este subproceso.

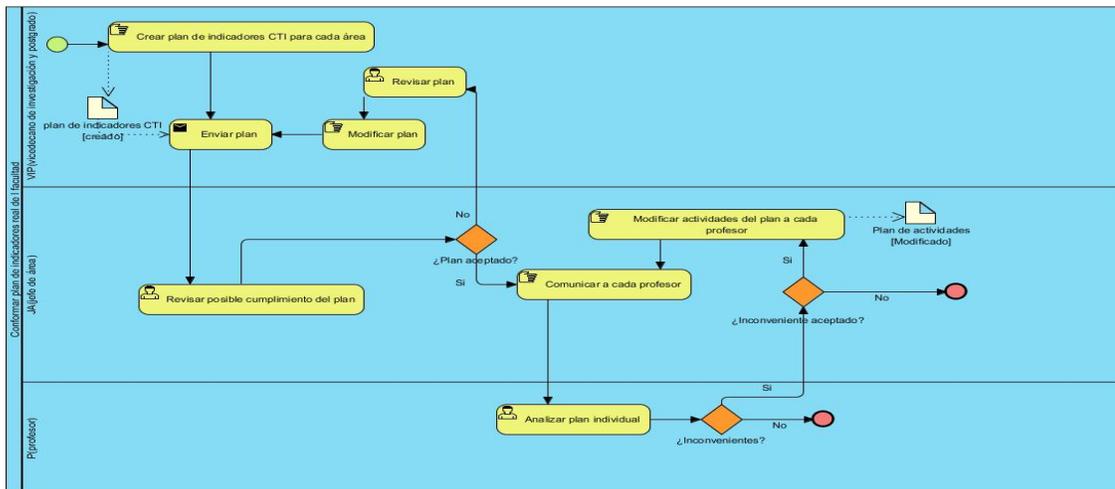


Figura 2.1. Conformar plan de indicadores real de la facultad

2.2.2. Subproceso 2: Actualizar indicadores y evidencia

Cuando un profesor o especialista obtiene algún resultado investigativo, éste hace llegar la evidencia generada al VIP1 o a su Jefe de Área (JA) personalmente o utilizando para ello el correo electrónico. Los cuales se encargan de depurar la información y en caso de que exista algún error se devuelve para arreglarlo. Cuando la evidencia se encuentra sin errores se procede a actualizar numéricamente el indicador en la herramienta informática que dispone la Universidad para esta tarea, el SIndiCIT. Luego se realiza la salva de las evidencias que respaldan cada uno de los indicadores almacenados. Para ello la dirección de Investigación y Postgrado UCI ha destinado un servidor ftp en la intranet local. La Figura 2.2 muestra el modelado de este subproceso.

¹Se define como áreas los departamentos y centros productivos de la facultad.

²Se define como Jefes de Área a los jefes de departamentos y subdirectores de centros productivos.

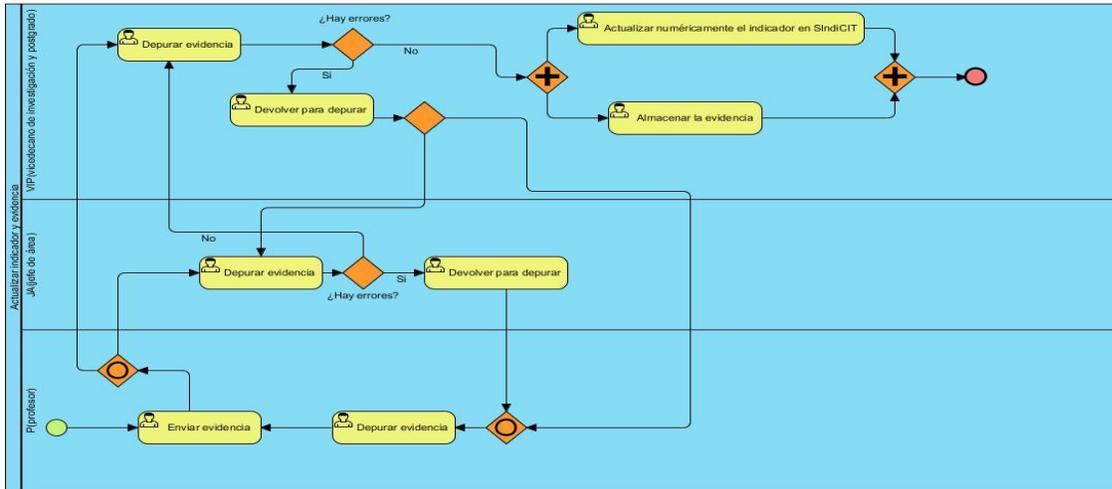


Figura 2.2. Actualizar indicadores y evidencia

2.2.3. Subproceso 3: Conformar Plan de Cursos

El JA se encarga de conformar el plan de cursos de cada área según su colectivo de trabajadores. Pone al tanto a sus profesores o especialistas y cuando recibe la aceptación de los mismos envía el plan de cursos al VIP1. La Figura 2.3 muestra el modelado de este subproceso.

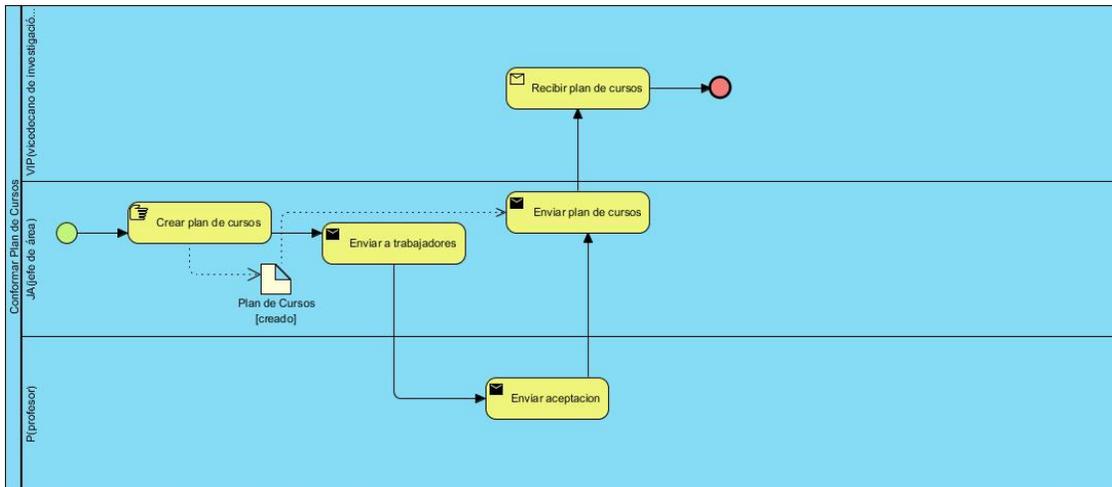


Figura 2.3. Conformar Plan de Cursos

2.2.4. Subproceso 4: Solicitud de impartir un curso

En la universidad se ofertan cursos de Postgrado, Maestría, Diplomado y Doctorado. Solo los profesores con categoría docente de Asistente o superior podrán solicitar impartir un curso. Cuando el profesor tiene los datos del curso presenta la solicitud a su JA o al VIP1 directamente. Si alguno de ellos encuentra errores en la depuración de la solicitud los envía a su subordinado respectivamente para corregir. Cuando la solicitud está completa se finaliza con la inscripción del curso, la notificación al profesor que impartirá el curso y la actualización numérica del indicador en la herramienta informática que dispone la Universidad para esta tarea, el SIndiCIT. La Figura 2.4 muestra el modelado de este subproceso.

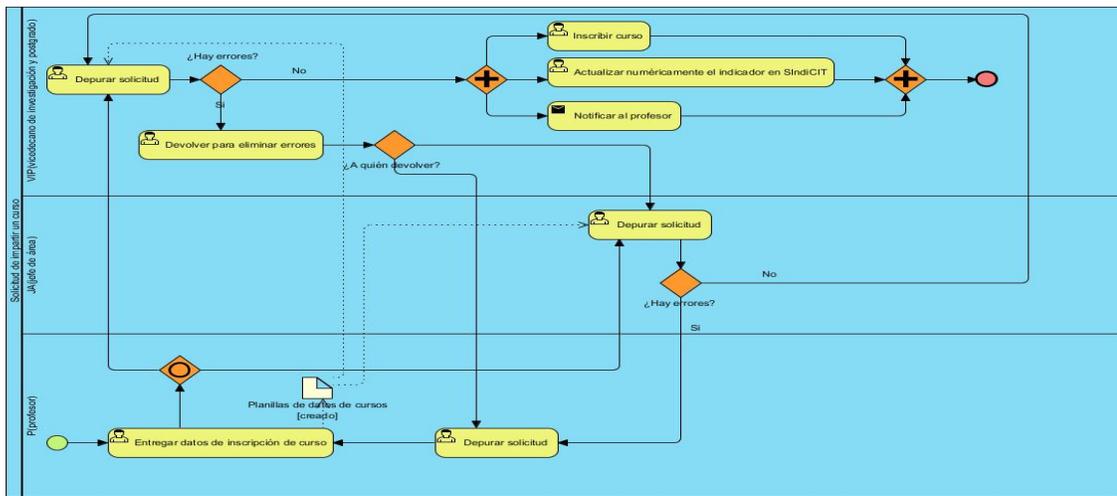


Figura 2.4. Solicitud de impartir un curso

2.2.5. Subproceso 5: Solicitud de matrícula a un curso

Cualquier Profesor (P) puede realizar la solicitud para matricular en un curso determinado. Si la matrícula no está llena, el Profesor que imparte Curso (PC) entrega la plantilla de inscripción para que el P solicite al JA su firma de autorizo. Cuando se emite el autorizo se le informa al VIP1 y el proceso concluye con la matrícula y notificación al P solicitante. La Figura 2.5 muestra el modelado de este subproceso.

Los 5 subprocesos antes descritos se realizan de forma semi-informatizada, lo que en ocasiones provoca atrasos, implica mucho más esfuerzo, tiempo y trae consigo desorganización, desactualización, ineficiente almacenamiento y la pérdida de datos.

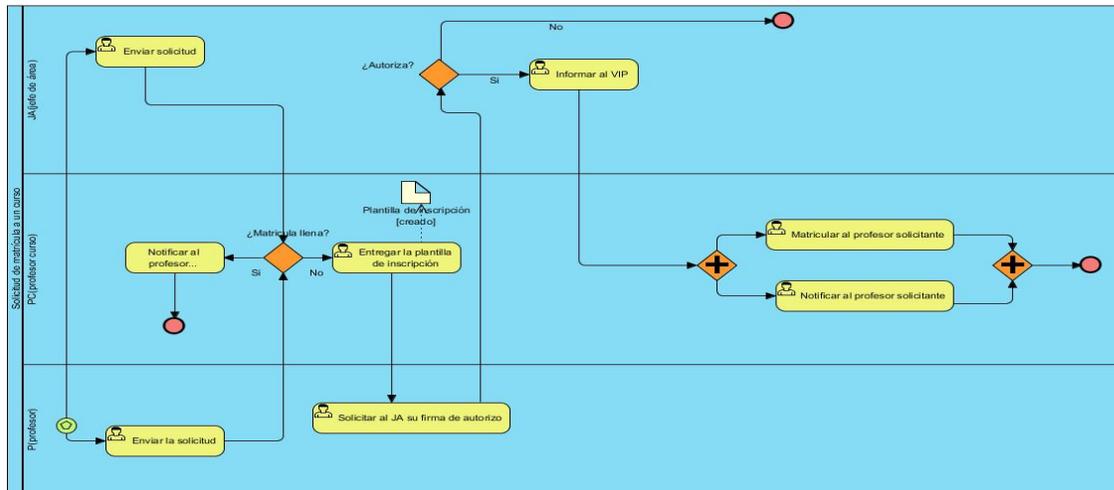


Figura 2.5. Solicitud de matrícula a un curso

2.3. Características de la Propuesta del Sistema

Con la implementación de la solución planteada se pretende informatizar el proceso de gestión de información de los indicadores de CTI y Postgrado en la facultad 2. Tendrán acceso a la aplicación solo el personal post-graduado de la facultad. Luego de que los usuarios se autenticuen, se le otorgarán los permisos según su rol para gestionar las evidencias, consultar información. Toda la información del sistema se guardará en una base de datos y los archivos digitales de las evidencias serán almacenadas en el mismo servidor donde se ejecute la aplicación; siguiendo la estructura de carpetas: año/trimestre/indicador. El sistema mostrará gráficas con los datos almacenados sobre el estado de los indicadores en la facultad permitiendo a los directivos conocer las áreas, la facultad en general y trabajadores mediante la consulta de evidencias por usuario con atrasos o sobre-cumplimiento de las actividades científicas planificadas.

Permitirá además la gestión de documentos útiles, la actualización de los indicadores de CTI y Postgrado, la publicación de anuncios, así como la generación de reportes o informes sobre la labor investigativa y post-graduada por cada departamento docente o centro productivo. También se manejará todo lo referente a la gestión de los profesores que tienen la categoría docente acorde para impartir un curso de postgrado así como la gestión de los profesores y especialistas que desean recibir dicho curso. Además el sistema permitirá consultar el ranking. En cuanto al diseño la aplicación se basará en la Estrategia Marcaria que establece la UCI. Será una aplicación web que permitirá un manejo adecuado y una eficiente gestión de toda la información referente a los procesos del VIP de la facultad 2.

Para mejor implementación del sistema en vez de trabajar por módulos se escogió desarrollarlo por roles como propone y trabaja el framework escogido. Uno de los puntos fundamentales que se debe tener en cuenta al desarrollar un sistema es la especificación de la audiencia a la que está dirigido el mismo, delimitando el nivel de interacción entre los diferentes usuarios. Los roles y las funciones que realizarán estos usuarios en

el sistema se detallan a continuación.

2.3.1. Roles y Funcionalidades

Vicedecano de Investigación y Postgrado (VIP1)

Además de gestionar toda la información del sistema es el encargado de la parte administrativa, y el trabajo con los roles y las áreas. El rol Vicedecano de Investigación y Postgrado VIP1 es el otorgado a las personas que se encargan de administrar el sistema en cuanto a las áreas, los roles e indicadores, tipos de indicadores, las categorías docentes del trabajador, evidencias y además es el rol que acepta y publica finalmente cualquier tipo de solicitud de usuarios con otros roles. Para cubrir cada una de las anteriores funciones se le permiten todos los permisos del rol P más los siguientes: Gestionar descargas, anuncios, los usuarios que acceden al menos una vez a la aplicación, las matrículas a cursos realizadas por los usuarios del sistema, las solicitudes de impartir cursos de los profesores, generar gráficas estadísticas e informes del estado de cumplimiento de los indicadores de CTI y Postgrado, se le posibilita asignar plan de indicadores por área, inscribir personalmente un curso que desee impartir o inscribir de manera personal a un profesor para que imparta un curso y administrar los resultados de indicador según los indicadores definidos y los tipos de indicadores correspondientes a esos indicadores.

Profesor (P)

Es el usuario básico. Es la persona que puede navegar por el sistema viendo toda la información publicada pero solo puede modificar sus datos. El rol Profesor P es el otorgado a todos los profesores y especialistas de la facultad 2, es un usuario básico que solo puede consultar algunas informaciones y realizar algunas solicitudes, es el rol por defecto de un usuario que accede por primera vez al sistema por supuesto si el usuario no es NI jefe de algún departamento docente, subdirector de algún centro productivo o el VIP1. Al mismo se le permite consultar el ranking con posibilidad de filtrar, documentos que son de interés para el área, cursos que se van a impartir en la facultad, anuncios de eventos y convocatorias, evidencias de los usuarios del sistema. Además se le permite gestionar sus solicitudes de evidencia, de anuncios que desea que publiquen, puede también matricularse en un curso que esté publicado cumpliendo con las condiciones que se establecen para matricularse e insertar solicitud de impartir un curso cumpliendo con los requerimientos que están establecidos para realizar dicha solicitud.

Jefe de Área (JA)

Profesor con el permiso para las gestiones del departamento que le corresponde. El rol Jefe de Área JA se le otorga a los Jefes de Departamento o Subdirectores de Centros de Desarrollo de la facultad. Pueden realizar ciertas funciones sobre su área y obtener información de su personal. Para lograr lo mencionado anteriormente se le permiten todos los permisos del rol P más los siguientes: Generar gráficas estadísticas, para mantener un control de los indicadores del área y sus niveles de cumplimiento; muestra un estado

general de los indicadores e informes del estado del cumplimiento de los indicadores, para tener siempre a la mano un resumen capaz de controlar los indicadores actuales del área.

Profesor que imparte Curso (PC)

Profesor con un curso asignado, que podrá gestionar dicho curso. El rol Profesor Curso PC se le otorga a los profesores que impartirán algún tipo de curso para que puedan modificar los aspectos relacionados con el mismo. Para realizar lo mencionado anteriormente se le permiten todos los permisos del rol P más los siguientes: Gestionar cursos, para controlar los cursos que impartirá y administrar las matrículas de cursos de los profesores en sus cursos.

2.4. Conclusiones del capítulo

En éste capítulo se analizó el flujo actual del proceso de gestión de información de los indicadores de CTI y Postgrado en el Vicedecanato de Investigación y PostgradoVIP de la Facultad 2, así como el personal que intervienen en el mismo, dígase personal: los roles que se definieron para que los usuarios se identifiquen con ellos y accedan a la información según su rol. Los roles se definieron a partir de las necesidades específicas del área. Se describen además los subprocesos que se originan y existen a partir del proceso llevado a cabo en esta área , se definieron las actividades que son objeto de informatización. Las cuales en conjunto con otra serie de funcionalidades propuestas para mejorar dicho negocio, conforman la lista de funcionalidades según el rol, que debe tener el sistema para dar solución al problema planteado. Además se exponen las características de la propuesta del sistema atendiendo al proceso que se maneja en el área y sus necesidades específicas.

Exploración, Planificación y Diseño del Sistema

3.1. Introducción

El desarrollo de aplicaciones es un proceso en el que se presentan artefactos que, luego de ser integrados, responden a las necesidades del cliente. El capítulo actual hace referencia a las fases de Exploración, Planificación y Diseño, propias de la metodología de desarrollo de software escogida. Además, se detallan los artefactos que propone para cada una de las fases mencionadas.

3.2. Exploración

La metodología de desarrollo de software XP comienza con la fase de exploración, en esta fase, los clientes plantean a grandes rasgos las Historias de Usuario Historia de Usuario (HU) como artefacto que se genera e identifica dicha fase. Las HU son porciones del comportamiento deseado de un sistema de software. Son muy utilizadas dentro del marco de desarrollo Ágil, y sirven para dividir una gran cantidad de funcionalidad en partes más pequeñas para facilitar la planificación.

Este concepto también se lo puede llamar “característica”, pero el término “historia” o “historia de usuario” se volvió muy popular dentro del marco Ágil. Kent Beck fue quien introdujo este término como parte de Extreme Programming para fomentar una manera informal y conversacional para la toma de requerimientos (en lugar de especificaciones largas escritas). De forma deliberada las historias no son detalladas hasta que están lista para ser desarrolladas; sólo se necesita el detalle necesario para permitir la priorización con otras historias(FOWLER, 2013).

Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se aprueba la tecnología y se exploran las posibilidades de la arquitectura del sistema. La fase de exploración toma de pocas semanas a pocos meses, dependiendo de la preparación y familiaridad que tengan los programadores con la tecnología (LETELIER y PENADÉS, 2006).

3.2.1. Historias de Usuario

Las HU sirven para registrar los requerimientos de los clientes según el negocio, y son utilizadas para poder realizar la estimación de cada una de las iteraciones durante la fase de planificación. Las HU son escritas por el equipo de trabajo en conjunto con los clientes en base a lo que se estima que es necesario para el sistema. Están escritas en un formato de oraciones en la terminología del cliente, sin necesidad de sintaxis técnicas. También son utilizadas para poder crear las pruebas de aceptación. Las HU solo proveen suficiente detalle para poder realizar la estimación de cuanto tardará en ser implementada dicha funcionalidad. Una gran diferencia entre las HU y los documentos tradicionales es que se centran en lo que el cliente necesita (BALAREZO, 2013).

En las HU se considera:

- La prioridad en el negocio:
 - Alta: Cuando son consideradas por los clientes esenciales para el funcionamiento del negocio.
 - Media: Cuando el cliente cree que son necesarias, pero estas no intervienen en gran medida en el desarrollo del negocio.
 - Baja: Cuando constituyen procesos que se deben tener en cuenta pero su ausencia no perjudica el flujo principal del negocio.
- El riesgo en desarrollo:
 - Alto: Cuando en la implementación de las HU pueden surgir errores que lleven a la inoperatividad del código.
 - Medio: Cuando en la implementación de las HU pueden existir errores que retrasen la entrega del producto.
 - Bajo: Cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del proyecto.
- Un Punto de Estimación equivale a una semana de programación, una semana de programación corresponde a 40 horas en desarrollo, 8 horas durante 5 días de la semana del calendario normal.
- Cuando se hace referencia a Todos como Usuario se refiere a P, VIP1, JA y PC.

A continuación las HU del sistema:

Tabla 3.1. Historia de usuario # 1

Historia de usuario	
Número: 1	Nombre: Autenticar usuario en el sistema
Usuario: Todos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	

Continúa en la próxima página

Tabla 3.1. Continuación de la página anterior

<p>Descripción: Antes de iniciar la aplicación se solicita el nombre de usuario y su clave para que tenga acceso a los datos que corresponden a su categoría de usuario. El sistema de no tener los roles definidos los crea de manera paralela cuando el usuario se loguea. Igual para con la asignación de áreas, el sistema le asigna al usuario de manera automática el área a la que corresponde dicho usuario. El sistema detecta y muestra un mensaje cuando hay problemas de conexión con el servidor UCI, y muestra un mensaje cuando hay campos vacíos o el usuario o contraseña es incorrecto.</p>
<p>Observaciones: Hay cuatro tipos de usuarios o roles, con distintos permisos de acceso al menú y funcionalidades que les corresponden. El usuario debe introducir usuario y contraseña UCI y debe ser profesor o especialista de la facultad 2.</p>

Tabla 3.2. Historia de usuario # 2

Historia de usuario	
Número: 2	Nombre: Gestionar Áreas
Usuario: VIP1	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alto
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Permite administrar las áreas con las que trabajará el sistema, posibilitando insertar área, eliminar área, listar áreas y editar áreas, ver los usuarios que pertenecen a un área, ver sus evidencias y datos de usuario y asignarle plan de Indicadores de CTI y Postgrado a las áreas. Además se muestra una gráfica de cómo están los indicadores a partir de las evidencias de los usuarios de la misma.	
Observaciones: Las áreas en la facultad son los Departamentos Docentes y los Centros de Desarrollo.	

Tabla 3.3. Historia de usuario # 3

Historia de usuario	
Número: 3	Nombre: Gestionar Roles
Usuario: VIP1	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se puede insertar roles, listar los roles, modificar un rol y eliminar un rol. Roles que tendrá establecido el sistema para ser asignados a los usuarios que accedan al sistema.	
Observaciones: Cada rol tendrá permisos diferentes según sus funciones.	

Tabla 3.4. Historia de usuario # 4

Historia de usuario	
Número: 4	Nombre: Gestionar Indicadores
Usuario: VIP1	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Permite al usuario gestionar los nombres de los indicadores que se guardan y controlan en el sistema pues los indicadores son la base de este sistema, pues las evidencias dependen de ello, las gráficas, los informes.	
Observaciones: Serán pocos los casos en que se tenga que añadir un nuevo indicador.	

Tabla 3.5. Historia de usuario # 5

Historia de usuario	
Número: 5	Nombre: Gestionar Tipos de Indicadores
Usuario: VIP1	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Dentro de cada indicador se podrá añadir, modificar o eliminar un tipo de indicador.	
Observaciones: Cada indicador tiene uno o varios tipos y para añadir un tipo debe haber creado previamente dicho indicador.	

Tabla 3.6. Historia de usuario # 6

Historia de usuario	
Número: 6	Nombre: Gestionar Resultado de Indicadores
Usuario: VIP1	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se podrán gestionar los resultados de indicadores ya definidos por la universidad, los mismos están asociados a un tipo de indicador correspondiente ya existente en el sistema.	
Observaciones: Los resultados no varían y están ya definidos. Se puede buscar los resultados de indicadores a partir de la selección del indicador, tipo de indicador y como resultado es la tabla con los resultados de ese filtro.	

Tabla 3.7. Historia de usuario # 7

Historia de usuario	
Número: 7	Nombre: Gestionar Categoría del Profesor

Continúa en la próxima página

3.2. EXPLORACIÓN

Tabla 3.7. Continuación de la página anterior

Usuario: VIP1	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se podrá gestionar las categorías docente ya definidas para los profesores en el sistema.	
Observaciones: Las categorías no varían y están ya definidas.	

Tabla 3.8. Historia de usuario # 8

Historia de usuario	
Número: 8	Nombre: Asignar Plan de Indicadores CTI por Área
Usuario: VIP1	
Prioridad en negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Permite al usuario: Asignarle un plan de Indicadores de CTI y Postgrado a las áreas deseadas. Plan que se muestra en las gráficas e informes.	
Observaciones: La cantidad de indicador no puede ser menor que cero. No se le puede asignar un plan a un área que no exista en el sistema.	

Tabla 3.9. Historia de usuario # 9

Historia de usuario	
Número: 9	Nombre: Gestionar Evidencias
Usuario: VIP1	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2.5	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Pueden añadirse nuevas evidencias de cualquier tipo de manera paralela el adjunto se guarda en la ruta evidencia/año/trimestre/indicador, a cada inserción de evidencia, se seleccionan los campos requeridos y se le añade un archivo adjunto que es la evidencia digital, ya sea una foto o una fotocopia de la original. Se pueden editar las evidencias y modificar sus datos siempre y cuando sean correctos. Al eliminar una evidencia se elimina el adjunto en la ruta anteriormente especificada.	
Observaciones: La evidencia introducida puede ser una nueva o una solicitada por un profesor P.	

Tabla 3.10. Historia de usuario # 10

Historia de usuario	
Número: 10	Nombre: Gestionar Usuarios

Continúa en la próxima página

Tabla 3.10. Continuación de la página anterior

Usuario: VIP1	
Prioridad en negocio: Alta	Riesgo en desarrollo: Alta
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Permite administrar los usuarios que están almacenados en el sistema. Cuando el VIP1 desea insertar un usuario, la inserción se realiza mediante un buscador que dado el usuario de la persona el sistema se basa en el servicio web de la UCI y obtiene los datos de ese usuario dado por hecho que el usuario tecleado sea correcto, sino no muestra resultado, cuando encuentra el usuario le puede asignar la categoría docente ,le asigna el área de manera automática como otro conjunto de datos, permite inscribir el usuario al boletín y asignarle el rol correspondiente al mismo.	
Observaciones: El usuario a insertar debe de pertenecer a la institución para realizar de manera exitosa la acción.	

Tabla 3.11. Historia de usuario # 11

Historia de usuario	
Número: 11	Nombre: Gestionar Anuncios
Usuario: VIP1	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se puede añadir un nuevo anuncio que al igual que cuando solicitan uno el adjunto se almacena en la ruta <i>evidencias/anuncios</i> o publicar un anuncio solicitado por otro rol como en la HU 6 de la tabla A.12, para informar al usuario de los eventos y cursos que se están ofertando, documentos de interés y otros, esto sería público en la página de portada. Mientras el VIP1 no edite la solicitud de anuncio y lo publique el anuncio no va a estar visible para los usuarios del sistema.	
Observaciones: Tendrán acceso a esta información todos los usuarios. A la vez que el VIP1 publica una solicitud de anuncio se muestra en la portada del sistema y en la funcionalidad <i>Mis Anuncios</i> que le aparecerán las solicitudes de anuncios aceptadas.	

Tabla 3.12. Historia de usuario # 12

Historia de usuario	
Número: 12	Nombre: Gestionar Cursos
Usuario: VIP1	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Permite Manejar todo lo referente a los cursos, y sus funcionalidades.	

Continúa en la próxima página

Tabla 3.12. Continuación de la página anterior

Observaciones: El mismo profesor PC podrá gestionar y modificar su curso y su matrícula pero quien aprueba todo es el VIP1.
--

Tabla 3.13. Historia de usuario # 13

Historia de usuario	
Número: 13	Nombre: Insertar Solicitud de Impartir un Curso
Usuario: PC	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1.5	Iteración asignada: 2
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Los datos introducidos deben de estar correctamente a la hora de realizar la solicitud de impartir un curso. A la vez que realiza la solicitud s almacena en la ruta evidencias/cursos el adjunto correspondiente. Luego de aceptada la solicitud de impartir un curso y se le realizó el cambio de rol a dicho profesor, al mismo se le habilita la funcionalidad <i>Mis Cursos</i> para que pueda gestionar todo lo referente a sus cursos.	
Observaciones: Solo los profesores de categoría docente de <i>Asistente</i> en adelante podrán solicitar impartir estos cursos.	

Tabla 3.14. Historia de usuario # 14

Historia de usuario	
Número: 14	Nombre: Gestionar Matrícula
Usuario: VIP1, PC	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El VIP1 tiene permisos para gestionar todo lo referente a las matrículas e incluso hasta asignarle a un profesor como matrícula de un curso. El PC nada más tiene permiso de gestionar las matrículas de los cursos que está impartiendo y en las que él es matrícula.	
Observaciones: No se puede editar una matrícula.	

Tabla 3.15. Historia de usuario # 15

Historia de usuario	
Número: 15	Nombre: Gestionar Solicitud de Evidencia
Usuario: Todos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	

Continúa en la próxima página

Tabla 3.15. Continuación de la página anterior

<p>Descripción: Se llenan los campos o modelos con los datos de la evidencia, adjunta la evidencia y envía la solicitud para que sea publicada a su nombre. Al hacer la solicitud el adjunto que subió como evidencia se almacena de manera paralela al envío. El usuario nada más puede editar las evidencias que él ha solicitado y el VIP1 las ha aprobado.</p>
<p>Observaciones: Si la evidencia solicitada tiene errores, el VIP1 es el encargado de mandarle una notificación al usuario para que la depure pero si está en sus manos puede rectificarla él mismo.</p>

Tabla 3.16. Historia de usuario # 16

Historia de usuario	
Número: 16	Nombre: Gestionar Solicitud de Matrícula
Usuario: Todos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
<p>Descripción: Permite al VIP1:</p> <ul style="list-style-type: none"> • Asignar profesor como matrícula a un curso si aún el curso su matrícula no está completa, si no tiene asignado ya ese curso y si aún está disponible de acuerdo a la fecha de cierre. • Eliminar usuario como matrícula a un curso paralelamente se decrementa la capacidad del curso. • Listar matrícula. • Ver datos de matrícula. • Buscar profesores que son matrícula de los cursos publicados, el sistema permite buscar por curso o usuario. 	
<p>Observaciones: El profesor P podrá matricularse o accediendo en la portada a <i>Cursos</i> y matricularse o accediendo a la funcionalidad <i>Ver más</i> e intentar matricularse en los cursos según los requisitos que plantean los mismo dígame: capacidad del curso no esté llena, no esté ya matriculado en dicho curso y esté dentro de la fecha de convocatoria .</p>	

Tabla 3.17. Historia de usuario # 17

Historia de usuario	
Número: 17	Nombre: Gestionar Solicitud de Curso
Usuario: Todos	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
<p>Descripción: Se puede solicitar la inscripción de impartir un curso, según la categoría del profesor que va a impartirlo. Cuando realiza la solicitud se realiza de manera paralela el almacenamiento del adjunto en la ruta evidencias/cursos. Podrá gestionar sus solicitudes de cursos, podrá gestionar las matrículas a sus cursos, puede ver todas las solicitudes de cursos aceptadas y buscar cursos ya sea por área, usuario o dentro de un rango de fecha.</p>	

Continúa en la próxima página

Tabla 3.17. Continuación de la página anterior

Observaciones: Solo los profesores de categoría docente <i>Asistente</i> o superior podrán solicitar impartir estos cursos.
--

Tabla 3.18. Historia de usuario # 18

Historia de usuario	
Número: 18	Nombre: Generar Gráficas Estadísticas
Usuario: VIP1, JA	
Prioridad en negocio: Media	Riesgo en desarrollo: Alta
Puntos estimados: 1.5	Iteración asignada: 2
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se puede generar gráficas del: <ul style="list-style-type: none"> • Estado general de los indicadores. • Estado de los indicadores por área. • Estado del indicador por los tipos de indicadores asociados. • Estado de los indicadores dentro de un rango de fecha para cada una de los filtros mencionados anteriormente. 	
Observaciones: Deben de estar bien definidos los indicadores, las áreas y tipos de indicadores para poder generar las gráficas pertinentes.	

Tabla 3.19. Historia de usuario # 19

Historia de usuario	
Número: 19	Nombre: Generar Informes del Estado de los Indicadores
Usuario: VIP1, JA	
Prioridad en negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Permite generar un informe del estado de los indicadores por: <ul style="list-style-type: none"> • Estado general de los indicadores de CTI y Postgrado. • Estado de los indicadores por área. • Estado general de los indicadores dentro de un rango de fecha. 	
Observaciones: El informe se puede filtrar por área y fecha además del general.	

Tabla 3.20. Historia de usuario # 20

Historia de usuario

Continúa en la próxima página

Tabla 3.20. Continuación de la página anterior

Número: 20	Nombre: Gestionar Descargas
Usuario: VIP1	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Posibilita subir, editar y eliminar del sistema los archivos de descarga que son de interés para el personal postgraduado de la facultad. Los cambios que realice en dicha funcionalidad se reflejan en la portada en la funcionalidad <i>Documentos del Área</i> . Cuando realiza la inserción se realiza de manera paralela el almacenamiento del adjunto en la ruta evidencias/descargas.	
Observaciones: Los archivos de descarga adjuntos podrán ser de cualquier formato.	

Tabla 3.21. Historia de usuario # 21

Historia de usuario	
Número: 21	Nombre: Gestionar Planes de Áreas
Usuario: VIP1	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se podrán gestionar los planes de área que ya están asignados, el campo año se insertar a la vez que se asigna un plan a un área esa asignación para la tabla de planes de área le asigna el año actual que es cuando se está realizando la asignación de indicadores a un área determinada. Se podrá realizar la búsqueda de planes escogiendo el área para entonces mostrar una tabla con la asignación de indicadores que tiene dicha área.	
Observaciones: A la hora de editar un plan, la cantidad de el indicador seleccionado no puede ser menor que cero.	

Tabla 3.22. Historia de usuario # 22

Historia de usuario	
Número: 22	Nombre: Gestionar Solicitud de Anuncios
Usuario: Todos	
Prioridad en negocio: Baja	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 3
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se puede solicitar la publicación de un anuncio de cualquier tipo, ya sea un curso, un evento, etc. Todos los usuarios pueden realizar solicitudes de anuncios que sean de interés al claustro post-graduado. Luego de aprobada la solicitud le aparece en sus solicitudes de anuncios al usuario correspondiente y en la portada del sistema.	
Observaciones: Solo el VIP1 aprueba dicha solicitud para que sea publicado el anuncio. Mientras no esté publicado no aparece en la portada ni en las solicitudes de anuncios de usuario correspondiente.	

Tabla 3.23. Historia de usuario # 23

Historia de usuario	
Número: 23	Nombre: Consultar Ranking
Usuario: Todos	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se obtiene un ranking de todos los usuarios del sistema, entre las áreas de la facultad, por área se obtiene el ranking entre los profesores o especialistas que pertenecen a la misma en dependencia de su cantidad de evidencias. Se puede filtrar por área, entre trabajadores y entre las áreas dentro de un intervalo de fechas especificado.	
Observaciones: Se hace a través de una puntuación donde cada evidencia es contada por igual.	

Tabla 3.24. Historia de usuario # 24

Historia de usuario	
Número: 24	Nombre: Consultar Descargas
Usuario: Todos	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se puede descargar cualquier archivo de la lista de descargas ofrecida.	
Observaciones: En la portada existirá una lista con los últimos 5 archivos subidos pero además aparecerá una opción <i>Ver más</i> que le permite al usuario ver todos los documentos de interés para el área que han sido publicados.	

Tabla 3.25. Historia de usuario # 25

Historia de usuario	
Número: 25	Nombre: Consultar Cursos
Usuario: Todos	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se pueden consultar los cursos que se están ofertando y matricularse si se desea y puede. Esta funcionalidad se encuentra en la portada del sistema a manera informativa.	
Observaciones: En la portada existirá una lista con los cursos que se van a impartir pero además aparecerá una opción <i>Ver más</i> que le permite al usuario ver todos los cursos que han sido publicados.	

Tabla 3.26. Historia de usuario # 26

Historia de usuario	
Número: 26	Nombre: Consultar Anuncios
Usuario: Todos	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se puede descargar cualquier archivo de la lista de descargas ofrecida.	
Observaciones: En la portada existirá una lista con los anuncios que han sido publicados que son de interés para los profesores y especialistas de la facultad pero además aparecerá una opción <i>Ver más</i> que le permite al usuario ver todos los anuncios que han sido publicados.	

Tabla 3.27. Historia de usuario # 27

Historia de usuario	
Número: 27	Nombre: Consultar Evidencias de Usuario
Usuario: Todos	
Prioridad en negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 0.5	Iteración asignada: 3
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Se puede ver las evidencias de cualquier usuario tanto por usuario, que por área que dentro de un rango de fecha y la combinación además de dichos filtros todo posibilitando una exactitud a la hora de consultar las evidencias de los usuarios en el sistema.	
Observaciones: El sistema debe de permitir buscar a las evidencias filtrándolas por área, fecha , usuario y la combinación de los mismos.	

3.3. Planificación

La metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores o gerentes. El proyecto comienza recopilando “Historias de usuarios”, las que sustituyen a los tradicionales “casos de uso”. Una vez obtenidas las “historias de usuarios”, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Si alguna de ellas tiene “riesgos” que no permiten establecer con certeza la complejidad del desarrollo, se realizan pequeños programas de prueba, para reducir estos riesgos. Una vez realizadas estas estimaciones, se organiza una reunión de planificación, con los diversos actores del proyecto (cliente, desarrolladores, gerentes), a los efectos de establecer un plan o cronograma de entregas en los que todos estén de acuerdo. Una vez acordado este cronograma, comienza una fase de iteraciones, en donde en cada una de ellas se desarrolla, prueba e instala unas pocas “historias de usuarios”(JOSKOWICZ, 2008).

En esta fase el cliente establece la prioridad de cada HU, de acuerdo con el valor que aporta para el

negocio. Los programadores estiman el esfuerzo asociado a cada HU. Se ordenan las HU según prioridad y esfuerzo, y se define el contenido de las entregas y/o iteraciones, apostando por enfrentar lo de más valor y riesgo cuanto antes. Ya en esta etapa tomando como base las acciones realizadas correspondientes a la etapa de exploración se traza el equipo de desarrollo una línea de tiempo para la hora de entregar un producto con la mayor calidad, eficiencia y eficacia posible. (LETELIER y PENADÉS, 2006).

3.3.1. Estimación de Esfuerzo

Las estimaciones de esfuerzo asociado a la implementación de las HU la establecen los programadores utilizando como medida el punto de estimación. Un punto de estimación equivale a una semana de programación. Las HU generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las HU que fueron terminadas en la última iteración (ibíd.).

Tabla 3.28. Estimación de esfuerzo por historia de usuario

Iteración	Historias de usuario	Puntos estimados (semanas)
1	1 Autenticar usuario en el sistema	0.5
	2 Gestionar Áreas	0.5
	3 Gestionar Roles	0.5
	4 Gestionar Indicadores	0.5
	5 Gestionar Tipos de Indicadores	0.5
	6 Gestionar Resultado de Indicadores	0.5
	7 Gestionar Categoría del Profesor	0.5
	8 Asignar Plan de Indicadores CTI por Área	1
	9 Gestionar Evidencia	2.5
	10 Gestionar Usuarios	2
	11 Gestionar Anuncios	1
2	12 Gestionar Cursos	1
	13 Insertar Solicitud de Impartir un Curso	1.5
	14 Gestionar Matrícula	1
	15 Gestionar Solicitud de Evidencia	1
	16 Gestionar Solicitud de Matrícula	1
	17 Gestionar Solicitud de Curso	1
	18 Generar Gráficas Estadísticas	1.5
	19 Generar Informes del Estado de los Indicadores	1
3	20 Gestionar Descargas	0.5
	21 Gestionar Planes de Áreas	0.5
	22 Gestionar Solicitud de Anuncios	1
	23 Consultar Ranking	0.5
	24 Consultar Descargas	0.5

Continúa en la próxima página

Tabla 3.28. Continuación de la página anterior

	25	Consultar Cursos	0.5
	26	Consultar Anuncios	0.5
	27	Consultar evidencias de usuario	0.5
Total			23.5

Es muy importante la estimación de esfuerzo tanto para el cliente como para el equipo de desarrollo pues así ya hay una guía por donde saber que se está realizando en el momento. A la hora de entrega de cada iteración dicha estimación de esfuerzo resulta de gran apoyo pues como están bien definidas las HU que se van a implementar en cada iteración y su tiempo de realización la calidad, la disciplina y el cumplimiento son los resultados de una buena puesta en práctica etapa de planificación dentro de la metodología escogida. Al hacer una estimación de esfuerzo por HU se concluye que las 27 HU agrupadas en 3 iteraciones poseen un tiempo de demora para su realización de 23.5 semanas.

3.3.2. Plan de Iteraciones

El proceso de desarrollo está compuesto por varias iteraciones. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las HU que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué HU se implementarán en cada iteración (para maximizar el valor de negocio). Al final de la última iteración el sistema estará listo para entrar en producción (ibíd.).

Iteración 1

En dicha iteración se implementarán las HU que les fueron asignado nivel de prioridad alta por el cliente por su impacto en el negocio, prioridad que indica que esas HU están relacionadas con la primera etapa del proceso de gestión de información de los indicadores de CTI y Postgrado en la facultad 2, creando con ellas el funcionamiento base del sistema. Además, se obtendrá una primera versión del producto en la que el cliente podrá probar algunas de las funcionalidades solicitadas como: Autenticar usuario en el sistema, Gestionar Áreas, Gestionar Roles, Gestionar Indicadores, Gestionar Tipos de Indicadores, Gestionar Resultado de Indicadores, Gestionar Categoría del Profesor, Asignar Plan de Indicadores CTI por Área, Gestionar Evidencia, Gestionar Usuarios y Gestionar Anuncios detalladas en las HU 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11.

Iteración 2

En esta iteración se implementarán las HU que les fueron asignado nivel de prioridad media por el cliente por su impacto en el negocio, prioridad que indica que esas HU están relacionadas con la segunda etapa del proceso de gestión de información de los indicadores de CTI y Postgrado en la facultad 2, creando con ellas un funcionamiento más abarcador del sistema. Además, se obtendrá una segunda versión del producto en

la que el cliente podrá probar algunas de las funcionalidades solicitadas como: Gestionar Cursos, Insertar Solicitud de Impartir un Curso, Gestionar Matrícula, Gestionar Solicitud de Evidencia, Gestionar Solicitud de Matrícula, Gestionar Solicitud de Curso, Generar Gráficas Estadísticas, Generar Informes del Estado de los Indicadores detalladas en las HU 12, 13, 14, 15, 16, 17, 18, 19.

Iteración 3

En dicha iteración se implementarán las HU que les fueron asignado nivel de prioridad bajo por el cliente por su impacto en el negocio, prioridad que indica que esas HU están relacionadas con la tercera etapa del proceso de gestión de información de los indicadores de CTI y Postgrado en la facultad 2, creando con ellas un funcionamiento completo y terminal del sistema. Además, se obtendrá la tercera y última versión del producto en la que el cliente podrá probar algunas de las funcionalidades solicitadas como: Gestionar Descargas, Gestionar Planes de Áreas, Gestionar Solicitud de Anuncios, Consultar Ranking, Consultar Descargas, Consultar Cursos, Consultar Anuncios, Consultar evidencias de usuario detalladas en las HU 20, 21, 22, 23, 24, 25, 26, 27.

Tabla 3.29. Plan de duración de las iteraciones

Iteración	Historias de usuario		Duración (semanas)
1	1	Autenticar usuario en el sistema	10.0
	2	Gestionar Áreas	
	3	Gestionar Roles	
	4	Gestionar Indicadores	
	5	Gestionar Tipos de Indicadores	
	6	Gestionar Resultado de Indicadores	
	7	Gestionar Categoría del Profesor	
	8	Asignar Plan de Indicadores CTI por Área	
	9	Gestionar Evidencia	
	10	Gestionar Usuarios	
	11	Gestionar Anuncios	
2	12	Gestionar Cursos	9.0
	13	Insertar Solicitud de Impartir un Curso	
	14	Gestionar Matrícula	
	15	Gestionar Solicitud de Evidencia	
	16	Gestionar Solicitud de Matrícula	
	17	Gestionar Solicitud de Curso	
	18	Generar Gráficas Estadísticas	
	19	Generar Informes del Estado de los Indicadores	
3	20	Gestionar Descargas	4.5
	21	Gestionar Planes de Áreas	
	22	Gestionar Solicitud de Anuncios	

Continúa en la próxima página

Tabla 3.29. Continuación de la página anterior

	23	Consultar Ranking	
	24	Consultar Descargas	
	25	Consultar Cursos	
	26	Consultar Anuncios	
	27	Consultar evidencias de usuario	
Total			23.5

3.3.3. Plan de Entrega

El plan de entrega, ver en la Tabla 3.30, es el resultado de tomar acuerdos con el cliente sobre el contenido de la primera entrega y determinar un cronograma para las demás entregas del producto (LETELIER y PENADÉS, 2006). Mediante el mismo el equipo de desarrollo mantiene al cliente informado y actualizado del progreso, funcionamiento y calidad del sistema. Gracias a el plan de entrega que se le confecciona a las iteraciones del sistema tanto el cliente como el equipo desarrollador se encuentran dirigidos hacia el avance y perfeccionamiento del producto. A continuación se refleja la evidencia de lo anteriormente planteado, donde se plasma la fecha en la que se va a realizar cada una de las entregas de las iteraciones del sistema por el equipo de desarrollo al cliente.

Tabla 3.30. Plan de Entregas

Iteración	Fecha de entrega
Iteración 1	17 de febrero de 2015
Iteración 2	21 de abril de 2015
Iteración 3	19 de mayo de 2015

3.4. Diseño

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente. Kent Beck dice que en cualquier momento el diseño adecuado para el software es aquel que: supera con éxito todas las pruebas, no tiene lógica duplicada, refleja claramente la intención de implementación de los programadores y tiene el menor número posible de clases y métodos (ibíd.).

3.4.1. Patrones de Arquitectura

Los patrones arquitectónicos abstraen los principales componentes de una arquitectura de sistema para ofrecer una solución a un problema. Por lo tanto, aplicar patrones arquitecturales permite al diseñador concentrarse en los detalles propios de su aplicación, utilizando soluciones ya existentes a los problemas recurrentes dentro de un dominio dado. También permite que su diseño sea más fácil de comprender por otros

participantes del proceso al utilizar una estructura bien identificada, facilitando las tareas de documentación, comunicación y potenciales extensiones o adaptaciones del sistema a nuevos requerimientos (BALLARI, 2001).

Los Patrones de Arquitectura definen la estructura básica de una aplicación. Pueden contener o estar contenidos en otros patrones. Proveen un subconjunto de subsistemas predefinidos, incluyendo reglas y pautas para su organización. Y son una plantilla de construcción (CARDOSO y CAMACHO, 2004).

Modelo Cliente Servidor

Se decide utilizar la modalidad o arquitectura Cliente-Servidor porque:

Es en la que confluyen una serie de aplicaciones basadas en dos categorías que cumplen funciones diferentes: una requiere servicios y la otra los brinda. Pero que a la vez, pueden realizar tanto actividades en forma conjunta como independientemente. En el caso del cliente es aquel que requiere un servicio del servidor. En esta categoría se realizan funciones de software basándose en el hardware, pero en caso de no tener la capacidad de procesar los datos necesarios, recurre al servidor y espera a que éste le brinde los servicios solicitados. El cliente es una estación de trabajo o computadora que está conectada a una red a través de la cual puede acceder al servidor.

Entre las características fundamentales de esta arquitectura encontramos que tanto el cliente como el servidor pueden realizar tareas en forma conjunta como separada ya que el cliente también tiene sus propias aplicaciones, archivos y bases de datos y que además, pueden estar en la misma plataforma o en plataformas diferentes. Por otra parte, el servidor puede brindar varios servicios a la vez, tanto al mismo cliente como a clientes múltiples (OCAMPO y MONTOYA, 2013).

Modelo Vista Controlador MVC

Se propone utilizar el patrón de arquitectura MVC ya que es un patrón de arquitectura de software que separa los datos, la interfaz y la lógica del sistema.

El Modelo: se encarga de la representación específica de toda la información con la cual el sistema va a trabajar; la lógica de datos puede llegar a asegurar la integridad de ellos y permitirá derivar nuevos datos. En el sistema se ve el uso del *modelo* como parte del patrón de arquitectura MVC pues ya symfony desde su instalación la estructura que trae por defecto es que en la carpeta *Form* en español formulario el programador guarde todo lo referente a las clases que se encargan del manejo con la base de datos. Dichas clases poseen su estándar particular de declaración que es el nombre de la clase que la primera letra es mayúscula acompañado de la palabra *Type* un ejemplo es: EvidenciaType.php.

La Vista: presenta el modelo con el que va a interactuar el usuario, más conocida como interfaz. En el sistema la *vista* al igual que el modelo ya vienen definidas dónde y cómo utilizarlas pues al instalar symfony el mismo tiene creado un ejemplo de un proyecto llamado Acme donde muestra la relación sencilla entre la modelo-vista-controller. Las vistas se emplearon en el sistema dentro del directorio *Resources/views/* se

crearon carpetas para cada controller. Dígase para el EvidenciaController existe una carpeta Evidencia que tiene en archivos .twig las vistas correspondientes a las acciones hechas en el controller.

El Controlador: responde a eventos, normalmente son acciones que el usuario invoca, implica cambios en el modelo y también en la vista (interfaz). En el sistema el *controlador* se evidencia al igual que las vistas y los modelos tiene una carpeta Controller que es donde se encuentran todas las clases controladoras del sistema. En dichas clases se realizan las acciones pertinentes que ejecuta el usuario desde las vistas el controlador consulta a la base de datos y responde lo correspondiente a dicha acción del usuario.

El patrón de arquitectura MVC cumple perfectamente con el fin particular de cualquier framework, una estructura bien definida que da soporte a un proyecto web, también ayuda a que el proyecto sea organizado y bien desarrollado. MVC es el patrón planteado por Symfony, framework que se usa para la solución propuesta en esta investigación. Como se muestra en la Figura 3.1, separa la vista de las aplicaciones de la lógica del negocio y del controlador. Divide una aplicación interactiva en tres componentes. El “modelo” contiene la información central y los datos. Las “vistas” despliegan información al usuario. Los “controladores” capturan la entrada del usuario. (ibíd.).

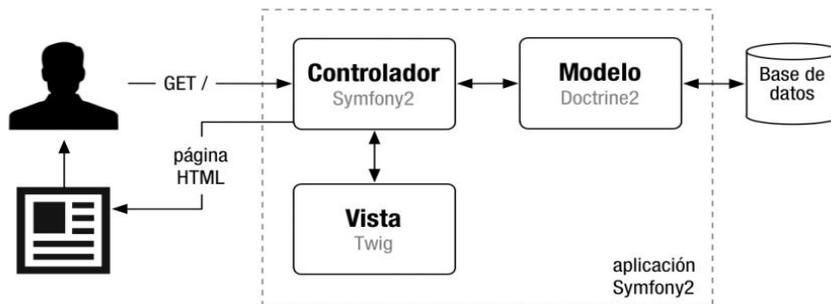


Figura 3.1. Diagrama del Modelo Vista Controlador usado por Symfony.

3.4.2. Patrones de Diseño

Un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular (F. BUSCHMANN y MEUNIER, 1996).

Patrones GRASP Los Patrones de Principios Generales para Asignar Responsabilidades (GRASP por sus siglas en inglés) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones.

Experto en Información: Las responsabilidades deben ser asignadas a las clases que poseen la información para realizar dicha responsabilidad. El Sistema de Gestión de Información para el Vicedecanato de Investigación y Postgrado (GIVIP) hace uso de este patrón y se evidencia cuando se desea mostrar todas las

evidencias almacenadas en el sistema, ya que la única clase con la responsabilidad de conocer esta información es Evidencia. De la misma manera sucede con las demás entidades del GIVIP. Solo la clase Usuario conoce quienes son los usuarios registrados en el sistema.

Creador: Asignarle a una clase la responsabilidad de crear una instancia de otra. Dentro del sistema este patrón se evidencia en las acciones de los controladores, las cuales crean objetos del modelo o los formularios que representan las entidades.

Alta Cohesión: Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Significa que las clases del sistema tienen asignadas solo las responsabilidades que les corresponde y mantienen una estrecha relación con el resto de las clases. Este patrón se garantiza en el sistema propuesto GIVIP, un ejemplo de ello se evidencia cuando se desea mostrar un gráfico sobre los indicadores de CTI y postgrado. La generación del gráfico es responsabilidad del GraficoController, pero quien tiene los datos a mostrar es la clase Indicadores. De esta manera se evidencia la relación que debe existir entre ambas clases ya que la primera solo se encarga de generar el gráfico, utilizando para ello, los datos que devuelven los métodos de Indicadores.

Bajo Acoplamiento: Determina el nivel de dependencia de una clase con respecto a otras. Una clase con bajo acoplamiento no depende de muchas otras. Este patrón es utilizado por el framework Symfony, y por ende en el sistema, al no asociar las clases del modelo con las de la vista o el controlador, la dependencia entre las clases, en este caso, se mantiene baja.

Controlador: Es el encargado de asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase que represente una de las siguientes opciones. Se evidencia el uso de este patrón en el GIVIP, ya que para cada petición o evento que se genere en el mismo, existe un controlador con la responsabilidad de obtenerla y devolver una respuesta. La respuesta puede ser mostrar una vista, ejecutar un método, devolver un mensaje, etc (F. BUSCHMANN y MEUNIER, 1996).

3.4.3. Modelo de Datos

Conjunto de herramientas conceptuales para describir la representación de la información en términos de datos. Los modelos de datos comprenden aspectos relacionados con: estructuras y tipos de datos, operaciones y restricciones (ZORRILLA, 2011).

El modelo de datos ver en la Figura 3.2 en esta investigación fue el punto de partida en aras de implementar un sistema de gestión de información que solucionara las necesidades específicas del VIP de la Facultad 2. Al elaborar el modelo de datos resultó de gran ayuda la utilización de herramientas informáticas que se han encargado de facilitarle el trabajo al hombre a la hora de confeccionar el mismo dentro de la que se encuentra la especificada en el Capítulo 1. Con el modelo de datos realizado, la lógica del negocio está representada a través de la relación entre las tablas y sus dependencias.

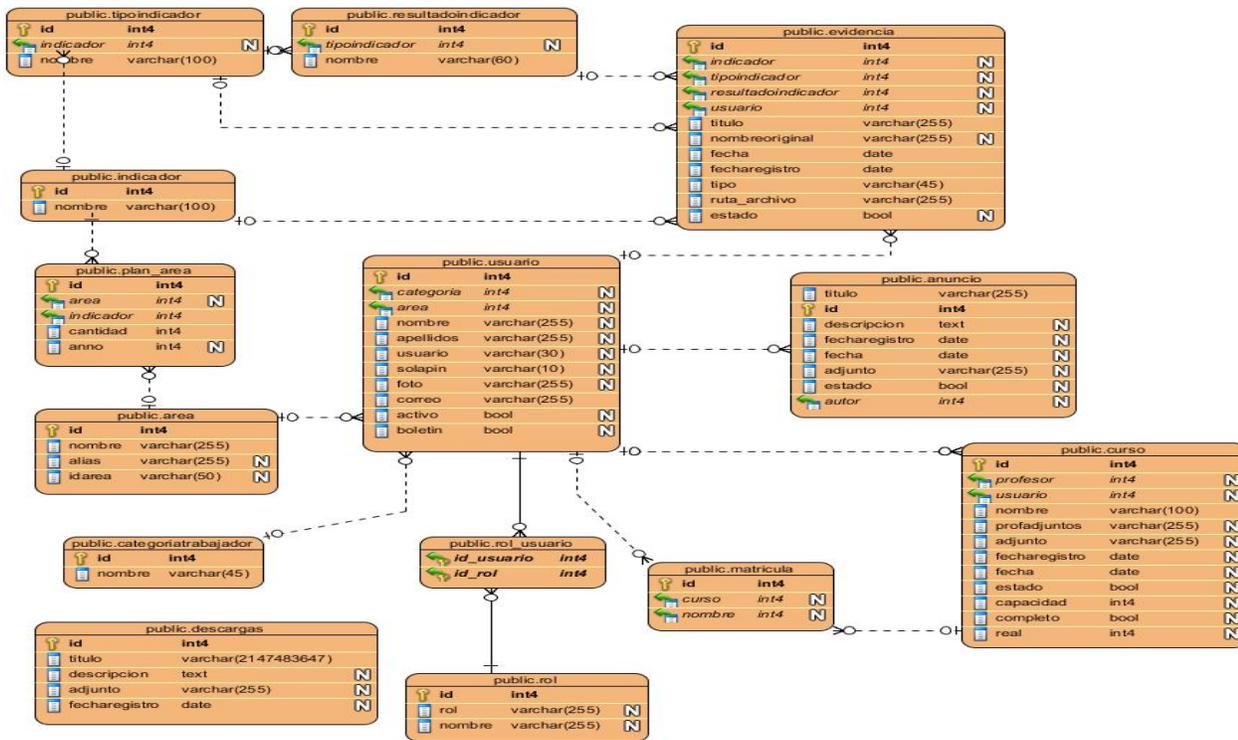


Figura 3.2. Diagrama Entidad Relación

3.4.4. Tarjetas Clase Responsabilidad Colaboración (CRC)

Las tarjetas CRC en la metodología XP se usan para para el diseño de software orientado a objetos. Estas tarjetas se dividen en tres secciones que contienen la información del nombre de la clase, sus responsabilidades y sus colaboradores. Una clase es cualquier persona, cosa, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades (BALAREZO, 2013).

Tabla 3.31. Tarjeta CRC # 1

Tarjeta CRC	
Clase: IndicadorController	
Responsabilidad	Colaboración
Esta clase es la encargada de realizar las acciones pertinentes para que el sistema pueda trabajar con los indicadores de manera correcta.	Indicador IndicadorType

Tabla 3.32. Tarjeta CRC # 2

Tarjeta CRC	
Clase: EvidenciaController	
Responsabilidad	Colaboración
Es la clase central del sistema y las más importante, la misma se encarga de gestionar las evidencias del sistema. Esta clase es la responsable de...	Evidencia EvidenciaType ResponsabilidadType

Tabla 3.33. Continuación de la página anterior

Esta clase se encarga de administrar todo lo referente a las áreas que posee el sistema, áreas en las que se agrupan los usuarios que accedan al sistema, además de asignarle plan indicadores a las áreas.	Area AreaType PlanificarType
---	------------------------------------

Tabla 3.34. Tarjeta CRC # 4

Tarjeta CRC	
Clase: UsuarioController	
Responsabilidad	Colaboración
En esta clase se realizan todas las acciones correspondientes a los usuarios que acceden al menos una vez al sistema.	Usuario UsuarioType BuscarusuarioType

Tabla 3.35. Tarjeta CRC # 5

Tarjeta CRC	
Clase: PlanAreaController	
Responsabilidad	Colaboración
Esta clase se encarga de a partir de la asignación de plan de indicadores a áreas realizar acciones sobre dichos planes.	PlanArea PlanAreaType BuscarplanareaType

Tabla 3.36. Tarjeta CRC # 6

Tarjeta CRC	
Clase: TipoindicadorController	
Responsabilidad	Colaboración
Esta clase gestiona a partir de los indicadores establecidos todo lo referente a los tipos de indicadores asociados a cada indicador.	Tipoindicador TipoindicadorType BuscartipoindicadorType

Tabla 3.37. Tarjeta CRC # 7

Tarjeta CRC	
Clase: RolController	
Responsabilidad	Colaboración
En esta clase es donde se definen los roles del sistema , roles que van ser asignados a los usuarios que accedan al sistema para realizar sus funciones en el sistema según su rol.	Rol RolType

Tabla 3.38. Tarjeta CRC # 8

Tarjeta CRC	
Clase: ResultadoindicadorController	
Responsabilidad	Colaboración
Esta clase a partir de tener los indicadores y los tipos de indicadores bien definidos gestiona todo lo que tiene que ver con los resultados de esos tipos de indicadores correspondientes a los indicadores.	Resultadoindicador ResultadoindicadorType BuscarresultadoindicadorType

Tabla 3.39. Tarjeta CRC # 9

Tarjeta CRC	
Clase: RankingController	
Responsabilidad	Colaboración
La clase permite que con la cantidad de evidencias que estén en el sistema, evidencias obtenidas por los usuarios mostrar un ranking entre los usuarios a nivel de Facultad, entre áreas, entre trabajadores de un área seleccionada y dentro de un rango de fecha en conjunto con los filtros anteriormente mencionados.	Evidencia

Tabla 3.40. Tarjeta CRC # 10

Tarjeta CRC	
Clase: MatriculaController	

Continúa en la próxima página

Tabla 3.40. Continuación de la página anterior

Responsabilidad	Colaboración
Esta clase se encarga de administrar los usuarios que desean matricularse en un curso, así como la gestión de las matrículas a cursos de los usuarios.	Matricula MatriculaType BuscarmatriculaType

El resto de las tarjetas CRC se encuentran en el Anexo B.

3.5. Conclusiones del capítulo

Se abordaron los artefactos que la metodología de desarrollo de software propone y que sirven de base para la próxima etapa de desarrollo. Se creó el Plan de Iteraciones agrupando las Historias de Usuario según su prioridad y se realizó el Plan de Entregas donde se estimó el tiempo de desarrollo en semanas para cada iteración. En este capítulo fueron analizados los patrones de diseño asociados a la arquitectura propuesta, los cuales darán mayor independencia a las clases y facilitarán la implementación. Se realizaron las tarjetas CRC para obtener una representación de las principales clases y funcionalidades del sistema. Como parte del diseño del sistema se realizó el diagrama entidad relación de la base de datos, permitiendo de esta forma almacenar los datos necesarios.

4.1. Introducción

La fase de implementación en el desarrollo de un producto de software, es el mecanismo donde se ponen en práctica todas las descripciones y arquitecturas propuestas en las fases de análisis y diseño, es el complemento del trabajo de las fases que lo preceden dentro del proceso de desarrollo de software. La implementación ofrece una materialización precisa de los requisitos. Una de las últimas fases del ciclo de vida antes de entregar un software para su explotación es la fase de pruebas, cuyo objetivo es comprobar si este cumple sus requisitos. Dentro de ella pueden desarrollarse varios tipos de pruebas en función de los objetivos de las mismas.

4.2. Implementación

En metodologías pesadas, la codificación es un proceso al cual solo se llega después de largas fases de análisis y diseño de las que queda una gran cantidad de documentación a partir de la cual el proceso de codificación es relativamente sencillo. En XP el proceso es muy diferente. Prácticamente desde un principio se inicia con la codificación, favoreciendo el logro del objetivo de estar haciendo entregas frecuentemente al cliente. Algunos de los elementos más importantes en cuanto a la codificación son que, el cliente siempre debe estar presente en esta, se debe trabajar en parejas y debe de haber una propiedad colectiva del código. Todos estos elementos representan paradigmas nuevos en lo que a la ingeniería de software se refiere, planteando entornos de discusión sobre la conveniencia de adoptarlas (ECHEVERRY y DELGADO, 2007).

En esta investigación se pusieron en práctica los conocimientos adquiridos del estudio sobre las metodologías ágiles y dentro de ellas XP. La misma está compuesta por fases y la implementación en conjunto con las pruebas juegan un papel intachable unidas a las demás. En ésta fase se realizó la implementación de las HU que fueron seleccionadas por cada iteración.

4.2.1. Código Fuente

Para obtener una versión funcional de la aplicación se deben implementar los componentes que se han definido, como resultado se obtienen archivos que contienen el código fuente de la aplicación. El código fuente de un software es un conjunto de líneas de texto que son las instrucciones que debe seguir la computadora para ejecutar dicho programa. Por tanto en el código fuente de un programa está escrito su funcionamiento. Estas instrucciones son escritas en un lenguaje de programación que consiste en un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones (LASSO, 2008).

Estándares de Codificación

XP enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados). Los estándares de programación mantienen el código legible para los miembros del equipo, facilitando los cambios (LETELIER y PENADÉS, 2006).

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, éste debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez (G. FERNÁNDEZ, 2005). Para facilitar el entendimiento del código y fijar un modelo a seguir, se establecieron estándares de codificación. A continuación se muestran algunos de estos estándares para el lenguaje PHP utilizados en Symfony2.

Nombres de clases y métodos

Para la definición de las clases y métodos en el código de la aplicación fue utilizado el estándar CamelCase. Este es un estilo de escritura que se aplica a frases o palabras compuestas. Existen dos tipos de estándares de CamelCase (ROGERS, 2014):

- UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula. Ejemplo: EjemploDeUpperCamelCase.
- lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula. Ejemplo: ejemploDeLowerCamelCase.

En esta investigación esta fue la notación con la que se implementó el código. De esta manera todas las clases en su nombre la primera letra de cada una de las palabras es mayúscula. Dicho estándar coincide con las pautas que establece PHP como lenguaje de programación utilizado para el desarrollo del sistema, resultando de gran beneficio para el equipo de desarrollo. A su vez se trabaja de manera homogénea basándose en dicho estándar; así como a la hora de crear un método que es igual que lo anterior con la excepción de que la primera letra es minúscula.

Código fuente 4.1. Ejemplo de código en PHP

```
1     ));
2
3     $form->add('submit', 'submit', array('label' => 'Buscar'));
4
5     return $form;
6 }
7
8 public function evidenciaXUsuarioAction($id)
9 {
10     $em = $this->getDoctrine()->getManager();
11
12     $usuario = $em->getRepository('GIVIPBundle:Usuario')->find($id);
13     if(!$usuario){
14         $this->get('session')->getFlashBag()->add('error', 'El usuario seleccionado
15             no existe en el sistema.');
```

```
16         return $this->redirect($this->generateUrl('evidencia'));
17     }
18     $nombre = $usuario->getNombre() ." ". $usuario->getApellidos();
19     if($this->get('security.context')->isGranted('ROLE_VIP1')){
20         $consulta = $em->createQuery("select e, i, u from GIVIPBundle:Evidencia e
21             JOIN e.indicador i JOIN e.usuario u WHERE e.usuario=:id
22             ");
23     }
24     else
25     {
26         $consulta = $em->createQuery("select e, i, u from GIVIPBundle:Evidencia e JOIN
27             e.indicador i JOIN e.usuario u WHERE e.usuario=:id and e.estado=true");
28     }
29
30     $consulta->setParameter('id', $id);
31
32     $res = $consulta->getResult();
33     $paginator = $this->get('ideup.simple_paginator');
34     $paginator->setItemsPerPage(10, 'resultado');
35     $entities = $paginator->paginate($res, 'resultado')->getResult();
36
37     $deleteForm = $this->createDeleteForm(0);
38     $buscarForm = $this->createSearchForm();
```

4.2.2. Tareas de Ingeniería

Las Tareas de Ingenierías son el artefacto que en la fase de implementación y prueba específicamente en la implementación encargada de albergar los detalles para la realización de cada una de las HU. Una HU

puede tener más de una tarea de ingeniería lo cual depende de la complejidad de dicha HU. Cuando el equipo de desarrollo conforma las tareas de ingeniería deja bien claro en sus campos a qué HU pertenece.

Los programadores responsables de implementar dicha funcionalidad, se establecen fecha de inicio y de fin en correspondencia con los datos que aporta la HU en cuestión, los puntos estimados que indican el tiempo real que se cogieron los programadores en realizar la funcionalidad, el tipo de tarea que se le va a realizar, nombre de la misma, y la parte final pero no menos importante la descripción de la tarea; en éste campo como su nombre lo indica se detallan todo lo que se va a realizar y en lo que se van a apoyar para darle solución a la funcionalidad correspondiente. Las Tareas de Ingeniería es el artefacto en el que se apoya el equipo desarrollador para dejar plasmado la manera en que se le dá solución a la HU a la que está asignada la tarea de ingeniería.

Tabla 4.1. Tarea de ingeniería # 1

Tarea	
Número de tarea: 1	Número de Historia de usuario: 1
Nombre de la tarea: Autenticar usuario en el sistema.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 1 de diciembre de 2014	Fecha de fin: 3 de diciembre de 2014
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: La presente tarea de ingeniería tiene como objetivo permitir que el usuario pueda introducir para autenticarse, su usuario y contraseña. Según sus privilegios basándonos en la consulta a ldap, accederá a los permisos correspondientes con su rol en la aplicación. Si está mal algún parámetro, debe salir un mensaje con el error correspondiente, sino irá a la página inicio. Además el sistema muestra un mensaje alertando si no hay conexiones de red con el servidor de la UCI.	

Tabla 4.2. Tarea de ingeniería # 2

Tarea	
Número de tarea: 2	Número de Historia de usuario: 2
Nombre de la tarea: Gestionar Áreas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 4 de diciembre de 2014	Fecha de fin: 7 de diciembre de 2014
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le va a permitir al VIP1 registrar las áreas con las que cuenta en ese momento la facultad dígase área a los departamentos docentes y los centros de producción, va a poder gestionarle los datos de los usuario que pertenecen a las áreas y están ubicados en ellas, ver además las gráficas estadísticas de cada área en particular y ver las evidencias de cada uno de los usuarios que pertenece a su correspondiente área.	

Tabla 4.3. Tarea de ingeniería # 3

Tarea	
Número de tarea: 3	Número de Historia de usuario: 3
Nombre de la tarea: Gestionar Roles.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 8 de diciembre de 2014	Fecha de fin: 11 de diciembre de 2014
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le va a permitir al VIP1 gestionar los roles que va a tener asignado los usuarios en el sistema. De que los roles no estén definidos el sistema lo hace de manera automática.	

Tabla 4.4. Tarea de ingeniería # 4

Tarea	
Número de tarea: 4	Número de Historia de usuario: 4
Nombre de la tarea: Gestionar Indicadores.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 7 de enero de 2015	Fecha de fin: 10 de enero de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le permitirá al VIP1 poder gestionar los indicadores. Indicadores que están establecidos a nivel de institución y es muy poco probable que varíen. Pero no obstante el sistema le dá la posibilidad de gestionarlos, pues de ellos surge todo lo demás referente a la actualización de los indicadores de CTI y Postgrado.	

Tabla 4.5. Tarea de ingeniería # 5

Tarea	
Número de tarea: 5	Número de Historia de usuario: 5
Nombre de la tarea: Gestionar Tipos de Indicadores.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 11 de enero de 2015	Fecha de fin: 14 de enero de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le permitirá al VIP1 luego que haya creado un indicador determinado adicionarle o gestionarle uno o más tipos de indicadores. Un mismo indicador no puede tener asignado dos tipos de indicadores iguales.	

Tabla 4.6. Tarea de ingeniería # 6

Tarea	
Número de tarea: 6	Número de Historia de usuario: 6

Continúa en la próxima página

Tabla 4.6. Continuación de la página anterior

Nombre de la tarea: Gestionar Resultado de Indicadores.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 19 de enero de 2015	Fecha de fin: 22 de enero de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le permitirá al VIP1 gestionar los resultados de indicadores. Teniendo como base los tipos de indicadores ya definidos en el sistema para entonces poder asignarle a un tipo e indicador los resultados de indicadores definidos. El sistema le permitirá realizar un búsqueda utilizando el filtro primeramente escoger el indicador, luego aparecen los tipos de indicadores correspondientes a ese indicador y luego el resultado es en una tabla los resultados correspondientes.	

Tabla 4.7. Tarea de ingeniería # 7

Tarea	
Número de tarea: 7	Número de Historia de usuario: 7
Nombre de la tarea: Gestionar Categoría del Profesor.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 15 de enero de 2015	Fecha de fin: 18 de enero de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le permitirá al VIP1 gestionar las categorías docentes de un profesor.	

Tabla 4.8. Tarea de ingeniería # 8

Tarea	
Número de tarea: 8	Número de Historia de usuario: 8
Nombre de la tarea: Asignar Plan de Indicadores CTI por Área.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 1 de enero de 2015	Fecha de fin: 6 de enero de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le va a permitir al VIP1 crear para cada área un plan de indicadores CTI a cumplir para poder lograr el plan que asignó la universidad para la facultad, además el sistema le va a permitir al vicedecano mostrar planificación de la facultad ,mostrar plan asignado para la facultad para que el claustro profesoral esté siempre informado, modificar plan y eliminar plan.	

Tabla 4.9. Tarea de ingeniería # 9

Tarea	
Número de tarea: 9	Número de Historia de usuario: 9

Continúa en la próxima página

Tabla 4.9. Continuación de la página anterior

Nombre de la tarea: Gestionar Evidencias.	
Tipo de tarea: Desarrollo	Puntos estimados: 2.5
Fecha de inicio: 12 de diciembre de 2014	Fecha de fin: 25 de diciembre de 2014
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le va a permitir al VIP1 depurar las solicitudes de evidencia y de estar correcta se actualizaría dicho indicador junto con su evidencia además de tener la posibilidad de editar o eliminar dicha evidencia, buscar evidencias dado un usuario, un área o dentro de un rango de fecha.	

Tabla 4.10. Tarea de ingeniería # 10

Tarea	
Número de tarea: 10	Número de Historia de usuario: 10
Nombre de la tarea: Gestionar Usuarios.	
Tipo de tarea: Desarrollo	Puntos estimados: 2
Fecha de inicio: 23 de enero de 2015	Fecha de fin: 3 de febrero de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le permitirá al VIP1 gestionar los usuarios del sistema. Cuando se edite un usuario posibilita hacerles cambios en el rol que tiene asignado, en la categoría docente, en el área que corresponde así como a los demás aspecto que tiene el usuario como si está o no suscrito al boletín. Esta funcionalidad es al igual que gestionar áreas y evidencias de vital importancia en el sistema.	

Tabla 4.11. Tarea de ingeniería # 11

Tarea	
Número de tarea: 11	Número de Historia de usuario: 11
Nombre de la tarea: Gestionar Anuncios.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 26 de diciembre de 2014	Fecha de fin: 31 de diciembre de 2014
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Luego que el profesor P envía al VIP1 la solicitud de un anuncio y este último lo depura(está correcto) el sistema le permite al vicedecano publicar dicho anuncio para que el claustro profesoral esté informado de todas las actividades que se están o se van a llevar a cabo en la institución. Además el sistema al vicedecano le permite publicar anuncios que él crea que sean de interés para los profesores.	

Tabla 4.12. Tarea de ingeniería # 12

Tarea	
Número de tarea: 12	Número de Historia de usuario: 12
Nombre de la tarea: Gestionar Cursos.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 1 de mayo de 2015	Fecha de fin: 6 de mayo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Luego que los cursos están publicados o no el sistema le permite al VIP1 modificar cursos, ver cursos, mostrar matrícula, mostrar solicitud de matrícula y eliminar matrícula, insertar a profesor como matrícula de un curso. Mientras que el profesor que va a impartir el curso si para realizar las acciones sobre sus cursos SÍ tienen que estar publicados dichos cursos.	

Tabla 4.13. Tarea de ingeniería # 13

Tarea	
Número de tarea: 13	Número de Historia de usuario: 13
Nombre de la tarea: Insertar Solicitud de Impartir un Curso.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha de inicio: 21 de marzo de 2015	Fecha de fin: 29 de marzo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Luego que el profesor P le envía la solicitud de impartir un curso al vicedecano, éste depura dicha solicitud y si está correcta la publica. Los datos cuando se realiza la inserción deben de estar correctos.	

Tabla 4.14. Tarea de ingeniería # 14

Tarea	
Número de tarea: 14	Número de Historia de usuario: 14
Nombre de la tarea: Gestionar Matrícula.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 30 de marzo de 2015	Fecha de fin: 5 de abril de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El VIP1 va a poder adicionar profesor P como matrícula a un curso así como ver, eliminar, y listar las matrículas.	

Tabla 4.15. Tarea de ingeniería # 15

Tarea	
Número de tarea: 15	Número de Historia de usuario: 15
Nombre de la tarea: Gestionar Solicitud de Evidencia.	

Continúa en la próxima página

Tabla 4.15. Continuación de la página anterior

Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 18 de febrero de 2015	Fecha de fin: 23 de febrero de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le va a permitir al P escoger de todos los indicadores que están en una lista no editable en el que el mismo participó además le va a permitir escoger dentro del indicador el tipo de ese indicador y dentro de este el resultado del indicador y como uno más de los campos la evidencia adjunta. Mediante el id del usuario que realiza la solicitud el sistema envía para sus solicitudes de evidencia las que son aceptadas, las solicitudes aparecen según el usuario que acceda a ellas.	

Tabla 4.16. Tarea de ingeniería # 16

Tarea	
Número de tarea: 16	Número de Historia de usuario: 16
Nombre de la tarea: Gestionar Solicitud de Matrícula.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 4 de marzo de 2015	Fecha de fin: 8 de marzo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Luego que el usuario ve los cursos de postgrado que están publicados en el sistema acciona sobre el que quiere matricularse y le aparece un botón que lleva por nombre Matricularse. Le aparece un mensaje informándole que se matriculó correctamente, que la matrícula está llena o que ya está matriculado en ese curso, luego de esa acción puede eliminar dicha matrícula, listar y ver dicha matrícula esto en el caso que sea satisfactorio.	

Tabla 4.17. Tarea de ingeniería # 17

Tarea	
Número de tarea: 17	Número de Historia de usuario: 17
Nombre de la tarea: Gestionar Solicitud de Curso.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 9 de marzo de 2015	Fecha de fin: 13 de marzo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El profesor si tiene categoría docente Asistente el sistema le va a permitir solicitar impartir un curso y al accionar sobre dicho botón le aparece un formulario con los campos pertinentes a llenar para realizar dicha solicitud, solicitud que le va a llegar al VIP1 y es decisión del mismo si se aprueba o no dicha solicitud de ser no es porque la solicitud posee errores y sería devuelta para que el P la depurara y enviarla corregida. De ser satisfactorio al usuario se le habilitaría la opción de <i>Mis Cursos</i> y a partir de ahí realizaría las gestiones que requiere impartir un curso.	

Tabla 4.18. Tarea de ingeniería # 18

Tarea	
Número de tarea: 18	Número de Historia de usuario: 18
Nombre de la tarea: Generar Gráficas Estadísticas.	
Tipo de tarea: Desarrollo	Puntos estimados: 1.5
Fecha de inicio: 14 de marzo de 2015	Fecha de fin: 22 de marzo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Al VIP1 el sistema le va a dar la posibilidad de generar gráficas estadísticas sobre el estado de los indicadores pero antes va a poder filtrar por área o por indicador o por fecha además del estado general de los indicadores.	

Tabla 4.19. Tarea de ingeniería # 19

Tarea	
Número de tarea: 19	Número de Historia de usuario: 19
Nombre de la tarea: Generar Informes del Estado de los Indicadores.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 15 de marzo de 2015	Fecha de fin: 20 de marzo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le va a permitir al VIP1 generar informes del estado de los indicadores, informes que se van a poder generar en el formato PDF, además va a poder filtrar o por área, o por fecha o por el estado general de los mismos.	

Tabla 4.20. Tarea de ingeniería # 20

Tarea	
Número de tarea: 20	Número de Historia de usuario: 20
Nombre de la tarea: Gestionar Descargas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 29 de abril de 2015	Fecha de fin: 31 de abril de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: Al VIP1 el sistema le va a dar la posibilidad de Gestionar las descargas que no es más que subir y eliminar archivos, documentos de interés para la comunidad post-graduada para mantenerla informada.	

Tabla 4.21. Tarea de ingeniería # 21

Tarea	
Número de tarea: 21	Número de Historia de usuario: 21
Nombre de la tarea: Gestionar Planes de Áreas	

Continúa en la próxima página

Tabla 4.21. Continuación de la página anterior

Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 11 de mayo de 2015	Fecha de fin: 14 de mayo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le permitirá al VIP1 gestionar los planes de áreas. Teniendo como base la asignación previa de los indicadores a las diferentes áreas.	

Tabla 4.22. Tarea de ingeniería # 22

Tarea	
Número de tarea: 22	Número de Historia de usuario: 22
Nombre de la tarea: Gestionar Solicitud de Anuncios.	
Tipo de tarea: Desarrollo	Puntos estimados: 1
Fecha de inicio: 28 de febrero de 2015	Fecha de fin: 4 de marzo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El usuario va a poder enviarle al VIP1 la solicitud de un anuncio que desea que se publique a manera informativa para la comunidad post-graduada, para ello el sistema le permitirá llenar un formulario que lleva por título Insertar Solicitud de Anuncios que en los campos del mismo están los detalles y requisitos que respecta a un Anuncio. Si se acepta la solicitud va a poder realizar las acciones correspondientes a la que le cambien el rol para PC.	

Tabla 4.23. Tarea de ingeniería # 23

Tarea	
Número de tarea: 23	Número de Historia de usuario: 23
Nombre de la tarea: Consultar Ranking.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 22 de abril de 2015	Fecha de fin: 25 de abril de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El usuario va poder filtrar para ver el ranking investigativo a nivel de área que corresponde el mismo o a nivel de facultad, estadística que va a mostrar como están los profesores, las áreas y los profesores dentro de un área específica en la esfera investigación y postgrado.	

Tabla 4.24. Tarea de ingeniería # 24

Tarea	
Número de tarea: 24	Número de Historia de usuario: 24
Nombre de la tarea: Consultar Descargas.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 25 de abril de 2015	Fecha de fin: 28 de abril de 2015

Continúa en la próxima página

Tabla 4.24. Continuación de la página anterior

Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva
Descripción: El usuario va a poder descargar los documentos de su interés que están publicados en el sistema. Al mismo le va a aparecer la opción mediante un botón que lleve por nombre Documentos del Área y al accionar sobre él va a poder descargar los de su interés, documentos que solamente los puede publicar el VIP1.

Tabla 4.25. Tarea de ingeniería # 25

Tarea	
Número de tarea: 25	Número de Historia de usuario: 25
Nombre de la tarea: Consultar Cursos.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 7 de mayo de 2015	Fecha de fin: 10 de mayo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le permitirá a todos los usuarios del sistema consultar los cursos que se están ofertando o se van a ofertar en la institución. Cursos que están publicados pues fueron aceptados por el VIP1 las solicitudes de cursos hechas por los profesores que contenían la información de interés y el conocimiento y dominio del tema que aborda el curso siendo de gran importancia para el claustro post-graduado.	

Tabla 4.26. Tarea de ingeniería # 26

Tarea	
Número de tarea: 26	Número de Historia de usuario: 26
Nombre de la tarea: Consultar Anuncios.	
Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 11 de mayo de 2015	Fecha de fin: 14 de mayo de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El sistema le permitirá a todos los usuarios del sistema consultar anuncios de convocatorias, eventos, concursos que se están efectuando o se van a efectuar en la institución. Anuncios que están publicados pues fueron aceptadas por el VIP1 las solicitudes de anuncios hechas por los profesores que contenían la información de interés para el claustro post-graduado.	

Tabla 4.27. Tarea de ingeniería # 27

Tarea	
Número de tarea: 27	Número de Historia de usuario: 27
Nombre de la tarea: Consultar Evidencias de Usuario.	

Continúa en la próxima página

Tabla 4.27. Continuación de la página anterior

Tipo de tarea: Desarrollo	Puntos estimados: 0.5
Fecha de inicio: 24 de febrero de 2015	Fecha de fin: 27 de febrero de 2015
Programador responsable: Osmar Capote Vázquez y Raúl Vázquez Leyva	
Descripción: El usuario mediante un botón que lleva por nombre <i>Servicios</i> que está ubicado en la parte superior izquierda, dentro del mismo está la funcionalidad de Buscar evidencias al accionar sobre dicha funcionalidad va a poder ver todas las evidencias en forma de lista del usuario deseado, basándose en filtros para hacer la búsqueda más precisa.	

4.3. Pruebas del Software

El proceso de producción del software requiere de pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual, debido a cambios durante esta fase (LETELIER y PENADÉS, 2006).

La automatización de las pruebas (CUKERMAN, 2006):

- Fomentan el cambio: facilitan la refactorización, puesto que permiten hacer pruebas sobre los cambios y así asegurarse de que los nuevos cambios no han introducido errores.
- Simplifica la integración: Puesto que permiten llegar a la fase de integración con un grado alto de seguridad de que el código está funcionando correctamente. De esta manera se facilitan las pruebas de integración.
- Documenta el código: Las propias pruebas son documentación del código puesto que ahí se puede ver cómo utilizarlo.

4.3.1. Pruebas Unitarias

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema (LETELIER y PENADÉS, 2006).

Son realizadas desde el punto de vista del programador y sirven, además de probar el código, para poder realizar refactorización del mismo. Cada programador, antes de comenzar a programar, debe preparar las pruebas unitarias. Esto hace que dichas pruebas estén preparadas para ser corridas durante la codificación y además, hace que al programador le surjan dudas y pueda evacuarlas con el cliente antes de empezar con la codificación. Una prueba unitaria es la verificación de un módulo (unidad de código) determinado dentro de un sistema. El concepto de “módulo” varía de acuerdo al lenguaje de programación que estemos utilizando; por ejemplo, en PHP sería una clase. Las pruebas unitarias nos aseguran que un determinado módulo cumpla con un comportamiento esperado en forma aislada antes de ser integrado al sistema. Los errores están más

acotados y son más fáciles de localizar: dado que se tienen pruebas unitarias que pueden desenmascararlos. (BALAREZO, 2013).

En la Figura 4.1 se muestra la interfaz del resultado de la aplicación de estas pruebas que se ejecutaron mediante PHPUnit, framework utilizado para las pruebas en PHP, es además el framework dentro de los que propone symfony que se escogió para la realización de dichas pruebas. Como se puede observar en la figura, se realizaron 11 test unitarios los cuales los 11 fueron exitosos.

```
C:\wamp\www\GIVIP2>php phpunit.phar -c app
PHPUnit 4.6.9 by Sebastian Bergmann and contributors.

Configuration read from C:\wamp\www\GIVIP2\app\phpunit.xml.dist

.....

Time: 349 ms, Memory: 30.25Mb

OK (11 tests, 30 assertions)
```

Figura 4.1. Resultados de la ejecución de los test unitarios

4.3.2. Pruebas de Aceptación

Los clientes escriben las pruebas funcionales para cada HU que deba validarse. Una HU no es aceptada hasta que haya pasado su prueba de aceptación. Las pruebas de aceptación representan algún tipo de resultado por parte del sistema. Los clientes son los responsables de verificar la exactitud de estas pruebas y de revisar los resultados para poder así priorizar las que fracasaron. Esto significa que en cada iteración se deben realizar nuevas pruebas de aceptación. Las pruebas de aceptación en XP permiten verificar que las funcionalidades de cada iteración se cumplan correctamente. El tester es el responsable de ayudar al cliente a seleccionar y escribir las pruebas de aceptación para cada historia de usuario. Tiene la responsabilidad de ayudar al cliente a tomar las decisiones correctas sobre que significa la calidad para su proyecto. (ibíd.).

Tabla 4.28. Prueba de aceptación # 1

Caso de prueba de aceptación	
Código: HU1_P1	Historia de usuario: 1
Nombre: Autenticar usuario en el sistema.	
Descripción: Prueba para la funcionalidad Autenticar usuario en el sistema.	
Condiciones de ejecución:	
El usuario tiene que ser profesor	
El usuario tiene que pertenecer a la Facultad 2	
El usuario tiene que pertenecer a un Departamento Docente o a un Centro de Producción	
El usuario y contraseña de la UCI deben ser válidos.	

Continúa en la próxima página

Tabla 4.28. Continuación de la página anterior

Pasos de ejecución: Se intenta autenticar un usuario en el sistema con los datos válidos.
Resultados esperados: El usuario se autentica correctamente en el sistema.

Tabla 4.29. Prueba de aceptación # 2

Caso de prueba de aceptación	
Código: HU2_P2	Historia de usuario: 2
Nombre: Gestionar Áreas.	
Descripción: Prueba para la funcionalidad Gestionar Áreas.	
Condiciones de ejecución: El VIP1 es el único que tiene los permisos para asignar plan de indicadores a cada área, para insertar una nueva área dentro de sus tres campos está <i>idarea</i> que debe de coincidir con el id que tiene esa área en el servicio web de la UCI, id del área que se desea insertar, para editar y eliminar un área. Los JA y el VIP1 son los que tienen el permiso de ver los usuarios que pertenecen al área seleccionada y ver una gráfica estadística de los indicadores, a partir de la evidencias subidas por los usuarios de dicha área. Los datos a la hora de insertar y editar un área deben ser correctos.	
Pasos de ejecución: El VIP1 accede a gestionar las áreas a las que pertenecen los profesores y especialistas.	
Resultados esperados: A la hora de gestionar un área todos los datos se manejan correctamente. Los usuarios se agrupan según al área que pertenecen , se pueden consultar sus evidencias, sus datos y se visualiza en un gráfica los indicadores actualizados según las evidencias de los usuarios.	

Tabla 4.30. Prueba de aceptación # 3

Caso de prueba de aceptación	
Código: HU3_P3	Historia de usuario: 3
Nombre: Gestionar Roles.	
Descripción: Prueba para la funcionalidad Gestionar Roles.	

Continúa en la próxima página

Tabla 4.30. Continuación de la página anterior

<p>Condiciones de ejecución: El VIP1 es el único que tiene permisos para administrar los roles. A la hora de insertar un rol en el sistema en el campo <i>Nombre</i> se pone el nombre que va a identificar el rol que pueden tomar los usuarios además en el campo <i>Nombre en el Sistema</i> se pone por ejemplo <i>ROLEPC</i> como lo establece el framework escogido para desarrollar el sistema informático . De que se eliminase algún rol establecido o no existiera, el sistema de manera automática lo crea paralelo al logueo de un usuario que requiere de dicho rol. El framework posee una archivo que lleva por nombre <i>segurity.yml</i> que es donde se establece la jerarquía de los roles en el sistema.</p>
<p>Pasos de ejecución: El VIP1 accede a la funcionalidad <i>Roles</i>, y tiene el permiso para administrarlos.</p>
<p>Resultados esperados: El VIP1 puede gestionar los roles del sistema correctamente.</p>

Tabla 4.31. Prueba de aceptación # 4

Caso de prueba de aceptación	
Código: HU4_P4	Historia de usuario: 4
Nombre: Gestionar Indicadores.	
Descripción: Prueba para la funcionalidad Gestionar Indicadores.	
<p>Condiciones de ejecución: El VIP1 es el único que tiene permiso de gestionar los indicadores Los indicadores deben de estar bien definidos.</p>	
<p>Pasos de ejecución: Accede a la funcionalidad <i>Indicadores</i> y realiza las acciones pertinentes para que a la hora de insertar un evidencia no hayan errores en el campo indicador.</p>	
Resultados esperados: El VIP1 gestiona los indicadores correctamente.	

Tabla 4.32. Prueba de aceptación # 5

Caso de prueba de aceptación	
Código: HU5_P5	Historia de usuario: 5
Nombre: Gestionar Tipos de Indicadores.	
Descripción: Prueba para la funcionalidad Gestionar Tipos de Indicadores.	

Continúa en la próxima página

Tabla 4.32. Continuación de la página anterior

<p>Condiciones de ejecución: El VIP1 es el único que tiene permiso de gestionar los tipos de indicadores Los indicadores deben de estar bien definidos. Los tipos de indicadores deben de estar bien definidos. Un mismo indicador no puede tener asignado dos tipos de indicadores iguales.</p>
<p>Pasos de ejecución: Accede a la funcionalidad <i>Tipo de Indicadores</i> y realiza las acciones pertinentes para que a la hora de insertar un evidencia no hayan errores en el campo tipo de indicador.</p>
<p>Resultados esperados: El VIP1 gestiona los tipos de indicadores correctamente.</p>

Tabla 4.33. Prueba de aceptación # 6

Caso de prueba de aceptación	
Código: HU6_P6	Historia de usuario: 6
Nombre: Gestionar Resultados de Indicadores.	
Descripción: Prueba para la funcionalidad Gestionar Resultados de Indicadores.	
<p>Condiciones de ejecución: El VIP1 es el único que tiene permiso de gestionar los resultados de indicadores. Los indicadores deben de estar bien definidos. Los tipos de indicadores deben de estar bien definidos. Los resultados de indicadores deben de estar bien definidos. Un mismo tipo de indicador no puede tener asignado dos resultados de indicadores iguales.</p>	
<p>Pasos de ejecución: Accede a la funcionalidad <i>Resultados de Indicadores</i> y realiza las acciones pertinentes para que a la hora de insertar un evidencia no hayan errores en el campo resultado de indicador.</p>	
Resultados esperados: El VIP1 gestiona los resultados de indicadores correctamente.	

Tabla 4.34. Prueba de aceptación # 7

Caso de prueba de aceptación	
Código: HU7_P7	Historia de usuario: 7
Nombre: Gestionar Categoría del Profesor.	
Descripción: Prueba para la funcionalidad Gestionar Categoría del Profesor.	
<p>Condiciones de ejecución: El usuario tiene que ser profesor. El usuario tiene que pertenecer a la Facultad 2. El usuario tiene que pertenecer a un Departamento Docente o a un Centro de Producción. El usuario y contraseña de la UCI deben ser válidos.</p>	

Continúa en la próxima página

Tabla 4.34. Continuación de la página anterior

<p>Pasos de ejecución: Se intenta autenticar un usuario en el sistema con los datos válidos.</p>
<p>Resultados esperados: El usuario se autentica correctamente en el sistema.</p>

Tabla 4.35. Prueba de aceptación # 8

Caso de prueba de aceptación	
Código: HU8_P8	Historia de usuario: 8
Nombre: Asignar Plan de Indicadores CTI por Área.	
Descripción: Prueba para la funcionalidad Asignar Plan de Indicadores CTI por Área.	
Condiciones de ejecución: El VIP1 es el único que tiene permiso de asignarle plan de indicadores por áreas.	
Pasos de ejecución: Accede a las áreas del sistema y acciona sobre la funcionalidad <i>asignar plan</i> .	
Resultados esperados: Se le asigna al área seleccionada el plan que le gestionó el VIP1.	

Tabla 4.36. Prueba de aceptación # 9

Caso de prueba de aceptación	
Código: HU9_P9	Historia de usuario: 9
Nombre: Gestionar Evidencia.	
Descripción: Prueba para la funcionalidad Gestionar Evidencia.	
Condiciones de ejecución: El VIP1 es el único encargado de gestionar las solicitudes de evidencias realizadas por los usuarios del sistema. Si no selecciona luego de un revisado los datos de las evidencias no le saldrán publicadas en sus solicitudes de evidencias ni en portada como últimas evidencias aceptadas en el sistema. A la hora de eliminar una evidencia se elimina simultáneamente el adjunto de la ruta en la que estaba almacenada Los datos a la hora de insertar y editar una evidencia deben ser correctos.	
Pasos de ejecución: Se realiza de manera simultánea cuando se inserta una solicitud de evidencia el adjunto en la ruta <i>evidencias/año/trimestre/indicador</i> .	
Resultados esperados: El VIP1 gestiona correctamente los datos de las evidencias en el sistema.	

Tabla 4.37. Prueba de aceptación # 10

Caso de prueba de aceptación	
Código: HU10_P10	Historia de usuario: 10
Nombre: Gestionar Usuarios.	
Descripción: Prueba para la funcionalidad Gestionar Usuarios.	
Condiciones de ejecución: El VIP1 es el único que tiene permiso de gestionar los usuarios del sistema. Los usuarios deben de existir en el servicio web que brinda la UCI . A la hora de editar un usuario primeramente debe de existir en la Base de Datos del sistema y además los datos que modifique deben de estar correctos. Un mismo usuario no puede estar en el sistema insertado más de una vez.	
Pasos de ejecución: Accede a la funcionalidad <i>Gestionar Usuarios</i> y realiza las acciones pertinentes para que a partir de todo lo que dependa el sistema del usuario que es en todo pues es la razón de ser del mismo pueda funcionar correctamente.	
Resultados esperados: El VIP1 gestiona los usuarios del sistema correctamente.	

El resto de las pruebas de aceptación se encuentran en el Anexo A.

A la propuesta de solución se le realizaron 11 test unitarios los cuales los 11 fueron exitosos. De los 27 casos de prueba de aceptación fueron resueltas las 4 no conformidades detectadas, y se realizaron los 7 cambios solicitados. Quedando aprobado por el cliente la propuesta de solución final planteada.

4.4. Conclusiones del capítulo

Se definió la notación estándar CamelCase para establecer una uniformidad a la hora de implementar el sistema. Las pruebas unitarias al igual que las pruebas de aceptación son una actividad fundamental en XP, las primeras deben ser automatizadas y elaboradas antes y durante la implementación de una clase, las segundas jugaron un papel primordial pues es la manera que tiene el cliente de verificar que las HU fueron implementadas correctamente. En éste capítulo se abordaron los temas referidos a la implementación y prueba de la investigación. Se cumplió con cada uno de los objetivos trazados para la validación y prueba del sistema como un todo ya que, cada uno de los resultados obtenidos en las diferentes iteraciones, resultaron de evaluación satisfactoria por las pruebas de aceptación aplicadas. Por tanto, la culminación del presente capítulo representa la garantía, calidad y satisfacción conjunta del cliente y el equipo de desarrollo con el producto final.

Conclusiones

Una vez realizado el desarrollo del sistema se arribaron las siguientes conclusiones:

1. Se realizó el estudio del estado del arte permitiendo identificar la necesidad de desarrollar un sistema para la Gestión de Información del Vicedecanato de Investigación y PostgradoVIP de la Facultad 2 puesto que ninguno de los estudiados cumplía con las necesidades específicas del área. Se justificaron las herramientas, metodología y tecnologías informáticas que se utilizaron en el desarrollo del trabajo las cuales posibilitaron la implementación del sistema en función de cumplir con el objetivo de la investigación.
2. Se describieron los procesos que se llevan en el área del VIP de la Facultad 2 permitiendo un mejor entendimiento del negocio ayudando esto a dar paso a una eficaz implementación del sistema.
3. Se implementaron las HU identificadas. Se realizaron las pruebas pertinentes definidas por la metodología de desarrollo de software XP para la validación de la solución propuesta. Para ello se utilizó el diseño y aplicación de las pruebas de aceptación y unitarias, validando la calidad de la solución propuesta con resultados positivos.

Atendiendo a lo anteriormente planteado se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos satisfactoriamente. La aplicación desarrollada contribuirá de manera significativa para la gestión de Información del Vicedecanato de Investigación y PostgradoVIP de la Facultad 2.

Recomendaciones

Los resultados obtenidos en este trabajo sugieren una serie de recomendaciones para seguir perfeccionando la gestión del proceso que realiza el Vicedecanato de Investigación y Postgrado VIP de la Facultad 2. Por lo que los autores recomiendan:

- El despliegue del sistema, para que el proceso de gestión de la información en el área se agilice y se realice de manera informatizada y centralizada.
- Analizar la posibilidad de extender el GIVIP a otras facultades.
- Perfeccionar la funcionalidad Ranking dándole una diferencia de valor a cada evidencia según su trascendencia y nivel (establecida por la dirección de investigación y postgrado de la UCI).

- BPMN** Modelado Business Process Modeling Notation. 7, 20, 22
- CES** Consejo de Educación Superior. 3
- CIDI** Centro de Ideoinformática. 5, 11
- CRC** Clase Responsabilidad Colaboración. 49, 52, 53
- CTI** Ciencia, Tecnología e Investigación. 3, 5, 11, 23, 27–29, 32, 34, 38, 42–44, 48, 58, 59, 71
- GEVIN** Sistema para la Gestión de Evidencias de Indicadores de CTI y Postgrado. 5, 11, 12
- GIVIP** Sistema de Gestión de Información para el Vicedecanato de Investigación y Postgrado. 47, 48, 74
- HU** Historia de Usuario. 30, 31, 35, 41–44, 54, 56, 57, 67, 72, 73
- JA** Jefe de Área. 24–26, 28, 31, 38, 68, 84, 85
- MES** Ministerio de Educación Superior. 3
- MVC** Modelo Vista Controlador. 22, 46, 47
- P** Profesor. 26, 28, 29, 31, 34, 37, 60–62, 81, 84
- PC** Profesor que imparte Curso. 26, 29, 31, 36, 64, 81, 84
- SGI** Sistema de Gestión de Información. 6, 9, 14, 22, 23
- SIndiCIT** Sistema de Indicadores Ciencia Tecnología e Innovación. 4, 11, 12, 24, 26
- UCI** Universidad de las Ciencias Informáticas. 2, 3, 11, 12, 24, 27, 32, 35, 57, 67, 68, 70, 72, 74, 90
- VIP** Vicedecanato de Investigación y Postgrado. 3, 5–7, 14, 22, 23, 27, 29, 48, 73, 74
- VIP1** Vicedecano de Investigación y Postgrado. 23–26, 28, 31–39, 57–65, 68–72, 80–82, 84–88
- XP** Programación Extrema. 14, 15, 22, 30, 41, 54, 67, 72, 73

Referencias bibliográficas

- AJA, Quiroga. 2002. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. *Acimed*. 2002, vol. 10, n.º 5, págs. 7-8.
- ALIAGA, Antonio. 2008. Servidores de Bases de Datos Postgresql con PgAdmin. 2008. Url: <https://iessanvicente.com/colaboraciones/postgreSQL.pdf>.
- ALLENDES, Rodolfo. 2015. Sistema de Gestión de la Investigación. *Universidad de Talca*. 2015. Url: http://sgi.otalca.cl/directorio_new.php.
- BALAREZO, Joel. 2013. Metodologías Ágiles, Programación Extrema XP. *Universidad Nacional de Trujillo*. 2013. Url: <http://es.slideshare.net/EvelingGiselleCruzVs/metodologia-monografia>.
- BALLARI, Tulio. 2001. Uso de patrones arquitectónicos web para el diseño de una aplicación domótica. En. *III Workshop de Investigadores en Ciencias de la Computación*. 2001. Url: <http://sedici.unlp.edu.ar/handle/10915/21731>.
- BUSCHMANN, F. y MEUNIER, R. 1996. Oriented Software Architecture. A System of Patterns. 1996.
- CARDOSO, E y CAMACHO, F. 2004. ARQUITECTURAS DE SOFTWARE. GUÍA DE ESTUDIO. 2004.
- CASTILLO, Alvaro. 2006. Webs dinámicas con PHP. 2006. Url: <http://www.programacion.net/php/tutorial/php4/2/>.
- CASTILLO, Francisco. 2015. Sistema Integral de Gestión Académica. *Universidad de Córdoba*. 2015. Url: <http://www.uco.es/servicios/informatica/corporativas/203-siga>.
- CHAÍN, Celia. 1999. Gestión de información para la investigación: Desarrollo y consolidación de las funciones del gestor como asesor y consultor para la ciencia y la tecnología. *Ciencias de la Información*. 1999, vol. 30, n.º 4, págs. 39-50.
- COELLO, Sayda. 2008. El paradigma cuantitativo de la investigación científica. *Ciudad de La Habana, Editorial Universitaria*. 2008. Url: <http://revistas.mes.edu.cu/elibro/libros/300/9789591603432.pdf/view>.
- CUBA Cuba. Convención, Constituyente. 1901. *Constitución de la República de Cuba*. 1901.
- CUKERMAN, Diego. 2006. Testing en eXtreme Programming. 2006.

- DE LOS ANGELES, Leodán. 2012. MARCOS DE TRABAJO PARA EL DESARROLLO DE APLICACIONES WEB. 2012. Url: http://repositorio_institucional.uci.cu/jspui/handle/ident/4139).
- DÍAZ, José Ramón. 2009. Las metodologías ágiles como garantía de calidad del software. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*. 2009, vol. 5, n.º 3, págs. 40-43.
- ECHEVERRY, Luís. y DELGADO, Elena. 2007. Caso práctico de la metodología ágil XP al desarrollo de software. 2007.
- EGUILUZ, Javier. 2009. Introducción a CSS. LibrosWeb. *Creative Commons*. 2009. Url: <http://www.librosweb.es/css/index.html>).
- ESPINOSA, Sam. 2009. Mapa de procesos. Business & Mgmt. 2009. Url: <http://www.slideshare.net/samespinosa/mapa-de-procesos.%201053479>).
- FERNÁNDEZ, Giovanni. 2005. Estándar codificación. *DOTNET*. 2005. Url: http://www.elguille.info/colabora/NET2005/giovannyfernandez_EstandarCodificacionNET.html).
- FERNÁNDEZ, H. 2009. Sistema Gestión De Bases De Datos. 2009. Url: <http://www.slideshare.net/hugofern/sistema-gestin-de-bases-de-datos>).
- FERNÁNDEZ, Viviana. 2008. La gestión de la información y las habilidades informacionales: binomio esencial en la formación universitaria. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal Sistema de Información Científica*. 2008, n.º 51, págs. 19-27.
- FOWLER, Martin. 2013. Las historias de usuario en pocas palabras. 2013.
- FUENTES, Eduardo. 2014. Investigación UTALCA. 2014. Url: <http://investigacion.atalca.cl/>).
- GAJDA, Włodzimierz. 2013. *Instant PhpStorm Starter*. 2013.
- GARCÍA, Miriam. 2014. Publicaciones Científicas en la Universidad de las Ciencias Informáticas. *Serie Científica de la Universidad de las Ciencias Informáticas*. 2014.
- GONZÁLEZ, Guillermo. 2014. Una introducción a Apache. 2014. Url: http://linux.ciberaula.com/articulo/linux_apache_intro).
- GUERRERO, Vera y JUDITH, Santa. 2010. Estudio de PHP y MySQL para el Desarrollo del Portal Web para el Municipio de Esmeraldas. 2010.
- HAMPTON, David. 1985. *Sistemas de Información*. 1985.
- HERNÁNDEZ, S. 2008. Metodología de la Investigación. *Editorial Félix Varela, La Habana*. 2008.
- HOGAN, Brian. 2011. HTML5 & CSS3. *Develop with Tomorrow's Standards Today*. 2011.
- JOSKOWICZ, José. 2008. Reglas y Prácticas en eXtreme Programming. 2008, págs. 10.
- LASSO, Ívan. 2008. Qué es el código fuente. 2008. Url: <http://www.proye%20ctoautodidacta.com/comics/que-es-el-codigo-fuente>).

- LETELIER, Patricio. y PENADÉS, Carmen. 2006. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). 2006. Url: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
- LLONCH, Jordi. 2013. The Book for Symfony 2.0. *Symfony*. 2013, vol. 3.0.
- MUSCIANO, Ch. y KEMEDY, B. 1999. HTML la guía completa. 2da edición. *México : McGRAW-HILL INTERAMERICANA EDITORES*. 1999.
- NÚÑEZ, Israel. y PÉREZ, Deborah. 2013. *DETERMINACIÓN DE LAS NECESIDADES DE FORMACIÓN E INFORMACIÓN UTILIZANDO EL MODELO AMIGA*. 2013.
- OCAMPO, Paola. y MONTTOYA, Adriana. 2013. Sistema de procesamiento de información de gestión digital de historias clínicas en entorno web (SPIGDATA_HC). 2013.
- ROGERS, Jessica. 2014. File Naming Standards for Digital Collections. 2014.
- SÁNCHEZ, Jordi. 2006. ¿Qué es un framework? 2006. Url: <http://jordisan.net/blog/2006/que-es-un-framework/>.
- SIERRA, María. 2006. Trabajando con Visual Paradigm for UML. 2006.
- UCI. 2011. Objetivos específicos de los procesos nivel cero UCI.doc. En. *Objetivos específicos de los procesos nivel cero UCI.doc*. 2011.
- UCI. 2013. *Procesos de la VRIP*. 2013.
- UCI. 2014. Sistema de Gestión Universitaria. *Universidad de las Ciencias Informáticas*. 2014. Url: <https://gestionuniversitaria.uci.cu/>.
- UCI. 2015. Dirección de Formación Postgraduada. *Universidad de las Ciencias Informáticas*. 2015. Url: <https://postgrado.uci.cu/news.php>.
- VEGA, John. 2011. Guía HTML5. El presente de la web. HTML5, css3 y javascript. *Un proyecto de Maestros del Web y Cristalab*. 2011, vol. 1.
- VENTURA, Toapanta. y ESTEFANÍA, Carolina. 2013. *Modelamiento, diseño de procesos bajo tecnología BPMN y propuesta de mejora en la empresa COCEBET SA*. 2013.
- WHITE, Stephen. 2004. "Introduction to BPMN". IBM Corporation. 2004. Url: <http://www.bpmn.org>.
- ZORRILLA, Marta. 2011. Modelos de datos. *Universidad de Cantabria*. 2011, págs. 2.

- AJA, Quiroga. 2002. Gestión de información, gestión del conocimiento y gestión de la calidad en las organizaciones. *Acimed*. 2002, vol. 10, n.º 5, págs. 7-8.
- ALBORNOZ, Mario. 2001. Política Científica y Tecnológica Una visión desde América Latina. 2001.
- ALIAGA, Antonio. 2008. Servidores de Bases de Datos Postgresql con PgAdmin. 2008. Url: <https://iessanvicente.com/colaboraciones/postgreSQL.pdf>.
- ALLENDES, Rodolfo. 2015. Sistema de Gestión de la Investigación. *Universidad de Talca*. 2015. Url: http://sgi.utalca.cl/directorio_new.php.
- BALAREZO, Joel. 2013. Metodologías Ágiles, Programación Extrema XP. *Universidad Nacional de Trujillo*. 2013. Url: <http://es.slideshare.net/EvelingGiselleCruzVs/metodologia-monografia>.
- BALLARI, Tulio. 2001. Uso de patrones arquitectónicos web para el diseño de una aplicación domótica. En: *III Workshop de Investigadores en Ciencias de la Computación*. 2001. Url: <http://sedici.unlp.edu.ar/handle/10915/21731>.
- BARCHINI, Graciela Elisa. 2005. *Métodos I + D de la Informática*. 2005.
- BECK, K. 1999. *Extreme Programming Explained. Embrace Change*. 1999.
- BUSCHMANN, F. y MEUNIER, R. 1996. *Oriented Software Architecture. A System of Patterns*. 1996.
- BUSCHMANN, Frank. y HENNEY, Kevlin. 2007. *PATTERN ORIENTED SOFTWARE ARCHITECTURE*. 2007.
- CALERO SOLÍS, Manuel. 2003. *Una explicación de la programación extrema*. 2003.
- CAMACHO, ERIKA.; CARDESO, FABIO. y NUÑEZ, GABRIEL. 2004. *ARQUITECTURAS DE SOFTWARE GUÍA DE ESTUDIO*. 2004.
- CARDOSO, E y CAMACHO, F. 2004. *ARQUITECTURAS DE SOFTWARE. GUÍA DE ESTUDIO*. 2004.
- CASTILLO, Alvaro. 2006. Webs dinámicas con PHP. 2006. Url: [http://www.programacion.net/php/tutorial/php4/2/.](http://www.programacion.net/php/tutorial/php4/2/)
- CASTILLO, Francisco. 2015. Sistema Integral de Gestión Académica. *Universidad de Córdoba*. 2015. Url: <http://www.uco.es/servicios/informatica/corporativas/203-siga>.

- CHAÍN, Celia. 1999. Gestión de información para la investigación: Desarrollo y consolidación de las funciones del gestor como asesor y consultor para la ciencia y la tecnología. *Ciencias de la Información*. 1999, vol. 30, n.º 4, págs. 39-50.
- COELLO, Sayda. 2008. El paradigma cuantitativo de la investigación científica. *Ciudad de La Habana, Editorial Universitaria*. 2008. Url: <http://revistas.mes.edu.cu/elibro/libros/300/9789591603432.pdf/view>.
- CUBA Cuba. Convención, Constituyente. 1901. *Constitución de la República de Cuba*. 1901.
- CUBA, República de. 2003. La estrategia de desarrollo científico y tecnológico en Cuba. 2003.
- CUKERMAN, Diego. 2006. Testing en eXtreme Programming. 2006.
- DE LOS ANGELES, Leodán. 2012. MARCOS DE TRABAJO PARA EL DESARROLLO DE APLICACIONES WEB. 2012. Url: http://repositorio_institucional.uci.cu/jspui/handle/ident/4139.
- DÍAZ, José Ramón. 2009. Las metodologías ágiles como garantía de calidad del software. *REICIS. Revista Española de Innovación, Calidad e Ingeniería del Software*. 2009, vol. 5, n.º 3, págs. 40-43.
- ECHEVERRY, Luís. y DELGADO, Elena. 2007. Caso práctico de la metodología ágil XP al desarrollo de software. 2007.
- EGUILUZ, Javier. 2009. Introducción a CSS. LibrosWeb. *Creative Commons*. 2009. Url: <http://www.librosweb.es/css/index.html>.
- ERLIJMAN PIWEN, Ariel. y GOYÉN FROS, Alejandro. 2001. *Problemas y Soluciones en la Implementación de Extreme Programing*. 2001.
- ESPINOSA, Sam. 2009. Mapa de procesos. Business & Mgmt. 2009. Url: <http://www.slideshare.net/samespinosa/mapa-de-procesos.%201053479>.
- FERNÁNDEZ, Giovanny. 2005. Estándar codificación. *DOTNET*. 2005. Url: http://www.elguille.info/colabora/NET2005/giovannyfernandez_EstandarCodificacionNET.html.
- FERNÁNDEZ, H. 2009. Sistema Gestión De Bases De Datos. 2009. Url: <http://www.slideshare.net/hugofern/sistema-gestin-de-bases-de-datos>.
- FERNÁNDEZ, Viviana. 2008. La gestión de la información y las habilidades informacionales: binomio esencial en la formación universitaria. *Red de Revistas Científicas de América Latina, el Caribe, España y Portugal Sistema de Información Científica*. 2008, n.º 51, págs. 19-27.
- FOWLER, Martin. 2013. Las historias de usuario en pocas palabras. 2013.
- FUENTES, Eduardo. 2014. Investigación UTALCA. 2014. Url: <http://investigacion.usalca.cl/>.
- GAJDA, Włodzimierz. 2013. *Instant PhpStorm Starter*. 2013.
- GARCÍA, Miriam. 2014. Publicaciones Científicas en la Universidad de las Ciencias Informáticas. *Serie Científica de la Universidad de las Ciencias Informáticas*. 2014.

- GAREA, B. y LANTIGUA, S. 2007. El Sistema de Programas y Proyectos de Ciencia, Tecnología e Innovación en Cuba. Contribución de las TIC en su gestión. 2007.
- GIANDINI, Roxana. y PÉREZ, Gabriela. 2010. Un lenguaje de Transformación específico para Modelos de Proceso del Negocio. En. *XXXVI Conferencia Latinoamericana de Informática (CLEI 2010)*. 2010.
- GONZÁLEZ, Guillermo. 2014. Una introducción a Apache. 2014. Url: http://linux.ciberaula.com/articulo/linux_apache_intro.
- GONZÁLEZ, Walter. y REYES, Oscar. 2012. Sistema para la gestión de la información del transporte en ETECSA. *Centro de Información y Gestión Tecnológica (CIGET) del CITMA. Filial del IDICT en Holguín*. 2012, vol. 18, n.º 1. Url: <http://www.ciencias.holguin.cu/index.php/cienciasholguin/article/view/634>.
- GUERRERO, Vera y JUDITH, Santa. 2010. Estudio de PHP y MySQL para el Desarrollo del Portal Web para el Municipio de Esmeraldas. 2010.
- HAMPTON, David. 1985. Sistemas de Información. 1985.
- HERNÁNDEZ, S. 2008. Metodología de la Investigación. *Editorial Félix Varela, La Habana*. 2008.
- HOGAN, Brian. 2011. HTML5 & CSS3. Develop with Tomorrow's Standards Today. 2011.
- IBARRA, Enrique José. 2002. POLÍTICA CIENTÍFICA NACIONAL. 2002.
- ICSU. 1999. Conferencia Mundial sobre la Ciencia para el Siglo XXI: Un nuevo compromiso. 1999.
- JOA MATOS, Loan. 2004. *Manual de Administración de Servidores de Bases de Datos Postgresql con PgAdmin*. 2004.
- JOSKOWICZ, José. 2008. Reglas y Prácticas en eXtreme Programming. 2008, págs. 10.
- LARMAN, Craig. 1999. *UML Y PATRONES INTRODUCCION AL ANALISIS Y DISEÑO ORIENTADO A OBJETOS*. 1999.
- LASSO, Ívan. 2008. Qué es el código fuente. 2008. Url: <http://www.proye%20ctoautodidacta.com/comics/que-es-el-codigo-fuente>.
- LEMARCHAND, Guillermo. 2010. *Sistemas Nacional de Ciencia, Tecnología e Innovación en América Latina y el Caribe*. 2010.
- LETELIER, Patricio. y PENADÉS, Carmen. 2006. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). 2006. Url: <http://www.cyta.com.ar/ta0502/v5n2a1.htm>.
- LLONCH, Jordi. 2013. The Book for Symfony 2.0. *Symfony*. 2013, vol. 3.0.
- MANSILLA. 2005. Indicadores de ciencia y tecnología para el desarrollo social. 2005.
- MENDOZA SANCHEZ, María. 2004. Metodologías De Desarrollo De Software. 2004.
- MINISTERIO DE CIENCIA, Tecnología y AMBIENTE, Medio. 2010. *ESTRATEGIA NACIONAL DE CIENCIA, TECNOLOGÍA E INNOVACIÓN*. 2010.

- MORENO, Nancy. y RODRÍGUEZ, Fermín. 2002. La gestión de la información como base a la gestión del conocimiento y del aprendizaje en las organizaciones en las universidades. *Educación Superior (Universidad de la Habana)*. 2002. Url: http://www.dict.uh.cu/Revistas/Educ_Sup/022002/Art030202.pdf).
- MUSCIANO, Ch. y KEMEDY, B. 1999. HTML la guía completa. 2da edición. México : McGRAW-HILL INTERAMERICANA EDITORES. 1999.
- NÚÑEZ, Israel. y PÉREZ, Deborah. 2013. *DETERMINACIÓN DE LAS NECESIDADES DE FORMACIÓN E INFORMACIÓN UTILIZANDO EL MODELO AMIGA*. 2013.
- OCAMPO, Paola. y MONTTOYA, Adriana. 2013. Sistema de procesamiento de información de gestión digital de historias clínicas en entorno web (SPIGDATA_HC). 2013.
- ONE. 2011. *CIENCIA Y TECNOLOGÍA en CIFRAS*. 2011.
- PASTRANA, Otero. y HERNÁNDEZ, Arbelo. 2012. "Sistema informático para la gestión de la información en las bibliotecas." Observatorio de la Economía Latinoamericana. 2012. Url: <http://www.eumed.net/coursecon/ecolat/cu/2012/opah.html>).
- PÉREZ, Martinto. y CARLOS, Pedro. 2001. El diseño metodológico de la investigación científica. Teoría de Muestreo: población y muestra. Diseño experimental y métodos. 2001.
- PÉREZ, X. y GUTIÉRREZ, Mario. 2010. Gestión de la información. *Universidad de León*. 2010. Url: <http://glossarium.bitrum.unileon.es/Home/gestion-de-la-informacion>).
- REYNOSO, Carlos. 2004. *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft*. 2004.
- REYNOSO, Carlos Billy. 2004. *Introducción a la Arquitectura de Software*. 2004.
- ROGERS, Jessica. 2014. File Naming Standards for Digital Collections. 2014.
- ROMERO, HILLER. y GRISEL, T. 2006. Indicadores de Ciencia y Tecnología para la toma de decisiones y la formulación de Políticas Públicas. Caso: Observatorio Nacional de Ciencia, Tecnología e Innovación (OCTI). 2006.
- SÁNCHEZ, Jordi. 2006. ¿Qué es un framework? 2006. Url: <http://jordisan.net/blog/2006/que-es-un-framework/>).
- SANTOVENIA, Javier. y TARRAGÓ, Consuelo. 2009. Sistemas de información geográfica para la gestión de la información. *Editorial Ciencias Médicas*. 2009, vol. 20, n.º 5. Url: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352009001100007).
- SENSIOLABS. 2014. *Identifying Design Patterns in the Symfony Framework*. 2014.
- SIERRA, María. 2006. Trabajando con Visual Paradigm for UML. 2006.
- SIMEÓN NEGRÍN, Rosa Elena. 2000. La ciencia y la tecnología en Cuba. En. *La ciencia y la tecnología en Cuba*. 2000.

- SINGLETON, Cedric. 2014. Sistema de Gestión de Información de Laboratorio (LIMS). *STARLIMS*. 2014.
- UCI. 2011. Objetivos específicos de los procesos nivel cero UCI.doc. En. *Objetivos específicos de los procesos nivel cero UCI.doc*. 2011.
- UCI. 2013. *Procesos de la VRIP*. 2013.
- UCI. 2014. Sistema de Gestión Universitaria. *Universidad de las Ciencias Informáticas*. 2014. Url: <https://gestionuniversitaria.uci.cu/>.
- UCI. 2015. Dirección de Formación Postgraduada. *Universidad de las Ciencias Informáticas*. 2015. Url: <https://postgrado.uci.cu/news.php>.
- VEGA, Iván de la. 2010. *Módulo de capacitación para la recolección y el análisis de indicadores de investigación y desarrollo*. 2010.
- VEGA, John. 2011. Guía HTML5. El presente de la web. HTML5, css3 y javascript. *Un proyecto de Maestros del Web y Cristalab*. 2011, vol. 1.
- VENTURA, Toapanta. y ESTEFANÍA, Carolina. 2013. *Modelamiento, diseño de procesos bajo tecnología BPMN y propuesta de mejora en la empresa COCEBET SA*. 2013.
- WHITE, Stephen. 2004. "Introduction to BPMN". IBM Corporation. 2004. Url: <http://www.bpmn.org>.
- ZORRILLA, Marta. 2011. Modelos de datos. *Universidad de Cantabria*. 2011, págs. 2.

Apéndices

A.1. Gestionar Anuncios

Tabla A.1. Prueba de aceptación # 11

Caso de prueba de aceptación	
Código: HU11_P11	Historia de usuario: 11
Nombre: Gestionar Anuncios.	
Descripción: Prueba para la funcionalidad Gestionar Anuncios.	
Condiciones de ejecución: El VIP1 es el único encargado de gestionar las solicitudes de anuncios realizadas por los usuarios del sistema. Si no selecciona luego de un revisado los datos de los anuncios no le saldrán publicados en sus solicitudes de anuncios ni en portada. A la hora de eliminar un anuncio se elimina simultáneamente el adjunto de la ruta en la que estaba almacenado Los datos a la hora de insertar y editar un anuncio deben ser correctos.	
Pasos de ejecución: Se realiza de manera simultánea cuando se inserta una solicitud de anuncio el adjunto en la ruta <i>evidencias/anuncio</i> .	
Resultados esperados: El VIP1 gestiona las solicitudes de anuncios correctamente.	

A.2. Gestionar Cursos

Tabla A.2. Prueba de aceptación # 12

Caso de prueba de aceptación	
Código: HU12_P12	Historia de usuario: 12

Continúa en la próxima página

Tabla A.2. Continuación de la página anterior

Nombre: Gestionar Cursos.
Descripción: Prueba para la funcionalidad Gestionar Cursos.
<p>Condiciones de ejecución:</p> <p>El VIP1 es el encargado de gestionar las solicitudes de cursos</p> <p>El VIP1 es el encargado de cambiarle al profesor que va a impartir el curso, curso que fue aceptado de P a PC, mientras no realice esta acción el usuario del curso no podrá gestionar los datos de su curso.</p> <p>A la hora de insertar o editar una solicitud los datos deben d estar correctos.</p>
<p>Pasos de ejecución:</p> <p>El usuario accede a las solicitudes de cursos.</p> <p>Cuando un usuario inserta una solicitud de impartir un cursos no se le muestra en sus cursos hasta que el VIP1 no los edita ,revisa y si está correcto lo publica</p> <p>El VIP1 accede a la funcionalidad <i>cursos</i> y gestiona las solicitudes que se han realizado.</p>
Resultados esperados: El VIP1 puede gestionar los cursos correctamente.

A.3. Insertar solicitud de impartir un curso

Tabla A.3. Prueba de aceptación # 13

Caso de prueba de aceptación	
Código: HU13_P13	Historia de usuario: 13
Nombre: Insertar solicitud de impartir un curso.	
Descripción: Prueba para la funcionalidad Insertar solicitud de impartir un curso.	
<p>Condiciones de ejecución:</p> <p>Los usuarios para hacer una solicitud de impartir un curso tienen que tener como mínimo de categoría docente <i>asistente</i>.</p> <p>El usuario va a poder eliminar, editar, ver sus solicitudes aceptadas por el VIP1, además de insertar solicitud de impartir un curso y ver la matrícula de los cursos luego de que le publiquen el curso y le cambien el rol de P a PC.</p> <p>Los datos a la hora de insertar una solicitud de curso deben ser correctos.</p>	
<p>Pasos de ejecución:</p> <p>El usuario accede a insertar solicitud de cursos.</p> <p>Cuando al usuario le aceptan la solicitud le se habilitan las funcionalidades correspondientes a su cambio de rol en el sistema.</p> <p>El usuario va a poder ver, eliminar y editar sus solicitudes que han sido publicadas, insertar una nueva solicitud de curso, además de consultar los cursos de otros usuarios y podrá ver todas las solicitudes de cursos aceptadas todo eso luego de que le acepten su solicitud y le cambien el rol.</p>	

Continúa en la próxima página

Tabla A.3. Continuación de la página anterior

Resultados esperados: El VIP1 gestiona correctamente las solicitudes de curso del profesor que desea impartir un curso.
--

A.4. Gestionar Matrícula

Tabla A.4. Prueba de aceptación # 14

Caso de prueba de aceptación	
Código: HU14_P14	Historia de usuario: 14
Nombre: Gestionar Matrícula.	
Descripción: Prueba para la funcionalidad Gestionar Matrícula.	
Condiciones de ejecución: Deben de haber cursos publicados. Los usuarios deben ser matrícula de un curso.	
Pasos de ejecución: Se accede a <i>matrícula</i> y se elimina, ve o inserta un usuario como matrícula a un cursos si tiene el rol para realizar esa acción que es el VIP1 quien inserta a un usuario como matrícula a un curso.	
Resultados esperados: El usuario gestiona las matrículas correctamente.	

A.5. Gestionar Solicitud de Evidencia

Tabla A.5. Prueba de aceptación # 15

Caso de prueba de aceptación	
Código: HU15_P15	Historia de usuario: 15
Nombre: Gestionar Solicitud de Evidencia.	
Descripción: Prueba para la funcionalidad Gestionar Solicitud de Evidencia.	
Condiciones de ejecución: El usuario va a poder solo editar y ver sus solicitudes aceptadas por el VIP1. El usuario tiene que hacer sus solicitudes de evidencia. Los datos a la hora de insertar y editar una solicitud de evidencia deben ser correctos.	
Pasos de ejecución: El usuario accede a sus solicitudes de evidencias. El usuario va a poder ver y editar sus solicitudes que han sido publicadas, insertar una nueva solicitud de evidencia, además de consultar las evidencias de otros usuarios y podrá ver todas las solicitudes de evidencias aceptadas.	

Continúa en la próxima página

Tabla A.5. Continuación de la página anterior

Resultados esperados: El usuario accede y gestiona sus solicitudes de evidencias que fueron aceptadas.

A.6. Gestionar Solicitud de Matrícula

Tabla A.6. Prueba de aceptación # 16

Caso de prueba de aceptación	
Código: HU16_P16	Historia de usuario: 16
Nombre: Gestionar Solicitud de Matrícula.	
Descripción: Prueba para la funcionalidad Gestionar Solicitud de Matrícula.	
Condiciones de ejecución: El usuario va a poder eliminar y ver sus solicitudes de matrícula a los cursos. El usuario puede matricularse en los cursos que aún la matrícula no está completa y aún está publicado dicho curso . El sistema no le permite al usuario matricularse en un mismo curso 2 veces en tiempo real.	
Pasos de ejecución: El usuario accede a sus solicitudes de matrículas a cursos. El usuario va a poder eliminar y ver sus solicitudes de matrículas a cursos, matricularse en los cursos que están publicados y aún no está su matrícula llena, además de consultar las matrículas de otros usuarios y podrá ver todas las matrículas a cursos hechas.	
Resultados esperados: El usuario accede y gestiona sus solicitudes de matrículas a cursos.	

A.7. Gestionar Solicitud de Curso

Tabla A.7. Prueba de aceptación # 17

Caso de prueba de aceptación	
Código: HU17_P17	Historia de usuario: 17
Nombre: Gestionar Solicitud de Curso.	
Descripción: Prueba para la funcionalidad Gestionar Solicitud de Curso.	

Continúa en la próxima página

Tabla A.7. Continuación de la página anterior

<p>Condiciones de ejecución:</p> <p>Los usuarios para hacer una solicitud de impartir un curso tienen que tener como mínimo de categoría docente <i>asistente</i>.</p> <p>El usuario va a poder eliminar, editar, ver sus solicitudes aceptadas por el VIP1 y ver, eliminar la matrícula de los cursos.</p> <p>El profesor podrá ver la funcionalidad correspondiente luego que el VIP1 le acepte la solicitud de impartir un curso y le cambia el rol dígase de P a PC.</p> <p>El usuario tiene que hacer sus solicitudes de impartir un curso.</p> <p>Los datos a la hora de insertar y editar una solicitud de curso deben ser correctos.</p>
<p>Pasos de ejecución:</p> <p>El usuario accede a sus solicitudes de cursos.</p> <p>Cuando un usuario se matricula en un curso de manera simultánea se incrementa la capacidad de matrícula del mismo al igual que cuando se des-matricula se decrementa.</p> <p>El usuario va a poder ver, eliminar y editar sus solicitudes que han sido publicadas, insertar una nueva solicitud de curso, además de consultar los cursos de otros usuarios y podrá ver todas las solicitudes de cursos aceptadas.</p>
<p>Resultados esperados: El usuario accede y gestiona sus solicitudes de curso que fueron aceptadas.</p>

A.8. Generar Gráficas Estadísticas

Tabla A.8. Prueba de aceptación # 18

Caso de prueba de aceptación	
Código: HU18_P18	Historia de usuario: 18
Nombre: Generar Gráficas Estadísticas.	
Descripción: Prueba para la funcionalidad Generar Gráficas Estadísticas.	
<p>Condiciones de ejecución:</p> <p>Deben de estar bien definidos los indicadores en el sistema</p> <p>Deben de estar bien definidas las áreas en el sistema</p> <p>Deben de estar bien definidos los tipos de indicadores en el sistema</p> <p>Para acceder a dicha funcionalidad el usuario debe de tener el rol de JA o VIP1.</p>	
<p>Pasos de ejecución:</p> <p>Acceder al enlace <i>Gráficas</i>.</p>	
<p>Resultados esperados: Al usuario se le generan las gráficas estadísticas</p>	

A.9. Generar Informes del Estado de los Indicadores

Tabla A.9. Prueba de aceptación # 19

Caso de prueba de aceptación	
Código: HU19_P19	Historia de usuario: 19
Nombre: Generar Informes del Estado de los Indicadores.	
Descripción: Prueba para la funcionalidad Generar Informes del Estado de los Indicadores.	
Condiciones de ejecución: Deben de estar bien definidos los indicadores en el sistema. Deben de estar bien definidas las áreas en el sistema. Para acceder a dicha funcionalidad el usuario debe de tener el rol de JA o VIP1.	
Pasos de ejecución: Acceder al enlace <i>Informes</i> .	
Resultados esperados: Al usuario se le generan los informes y puede exportarlos a PDF.	

A.10. Gestionar Descargas

Tabla A.10. Prueba de aceptación # 20

Caso de prueba de aceptación	
Código: HU20_P20	Historia de usuario: 20
Nombre: Gestionar Descargas.	
Descripción: Prueba para la funcionalidad Gestionar Descargas.	
Condiciones de ejecución: El usuario tiene que tener el rol de VIP1. Los datos a la hora de insertar y editar un documento de interés para el área deben ser correctos.	
Pasos de ejecución: El VIP1 accede a gestionar los documentos que son de interés para el claustro post-graduado. Al insertar un documento el mismo de manera simultánea se muestra en Documentos del Área en la portada del sistema y se guarda el adjunto en la ruta evidencias/descargas .	
Resultados esperados: Los documentos son publicados correctamente en la portada del sistema.	

A.11. Gestionar Planes de Área

Tabla A.11. Prueba de aceptación # 21

Caso de prueba de aceptación	
Código: HU21_P21	Historia de usuario: 21
Nombre: Gestionar Planes de Área.	

Continúa en la próxima página

Tabla A.11. Continuación de la página anterior

Descripción: Prueba para la funcionalidad Gestionar Planes de Área.
Condiciones de ejecución: El VIP1 es el único que tiene permiso de gestionar los planes de área. Los indicadores deben de estar bien definidos. Las áreas deben de tener asignaciones de plan de indicadores. La cantidad del indicador seleccionado no puede ser negativa. A la hora de editar una asignación de plan de indicadores los datos que se modificaron deben de ser correctos.
Pasos de ejecución: Accede a la funcionalidad <i>Planes de Área</i> y realiza las acciones pertinentes para tener un control total de los planes que han sido realizados.
Resultados esperados: El VIP1 gestiona los planes de área correctamente.

A.12. Gestionar Solicitud de Anuncios

Tabla A.12. Prueba de aceptación # 22

Caso de prueba de aceptación	
Código: HU22_P22	Historia de usuario: 22
Nombre: Gestionar Solicitud de Anuncios.	
Descripción: Prueba para la funcionalidad Gestionar Solicitud de Anuncios.	
Condiciones de ejecución: El usuario va a poder solo editar y ver sus solicitudes aceptadas por el VIP1. El usuario tiene que hacer sus solicitudes de anuncios. Los datos a la hora de insertar y editar una solicitud de anuncio deben ser correctos.	
Pasos de ejecución: El usuario accede a sus solicitudes de anuncios. El usuario va a poder ver y editar sus solicitudes que han sido publicadas, insertar una nueva solicitud de anuncio, además de consultar los anuncios de otros usuarios y podrá ver todas las solicitudes de anuncios aceptadas.	
Resultados esperados: El usuario accede y gestiona sus solicitudes de anuncios que fueron aceptadas.	

A.13. Consultar Ranking

Tabla A.13. Prueba de aceptación # 23

Caso de prueba de aceptación	
Código: HU23_P23	Historia de usuario: 23
Nombre: Consultar Ranking.	
Descripción: Prueba para la funcionalidad Consultar Ranking.	
Condiciones de ejecución: El usuario se autentique correctamente. El usuario para ver el ranking entre trabajadores dentro de un intervalo de fecha, la fecha inicio debe de ser menor que la fecha fin . El usuario para ver el ranking entre trabajadores dentro de un intervalo de fecha, la fecha inicio y la fecha fin no pueden estar vacías. El mismo mecanismo se practica para el ranking entre áreas y ranking entre trabajadores dentro de un área seleccionada dentro de un rango de fecha.	
Pasos de ejecución: Consulta el ranking manipulando los datos correctamente.	
Resultados esperados: El usuario se informa en el estado que se encuentra su área y de más áreas, como trabajador de manera individual en la Facultad y cómo están los trabajadores dentro de sus áreas consultando el ranking correctamente en el sistema.	

A.14. Consultar Descargas

Tabla A.14. Prueba de aceptación # 24

Caso de prueba de aceptación	
Código: HU24_P24	Historia de usuario: 24
Nombre: Consultar Descargas.	
Descripción: Prueba para la funcionalidad Consultar Descargas.	
Condiciones de ejecución: El VIP1 es el encargado de publicar los documentos de interés para el área. El sistema le permite a los usuarios descargar o ver los documentos publicados.	
Pasos de ejecución: En la portada del sistema le aparece la opción Documentos del Área. El usuario accede a la misma y se informa mediante dicha información publicada. Los usuarios podrán acceder a todos los documentos que se han publicado en el sistema.	
Resultados esperados: El usuario consulta los documentos de interés en el sistema.	

A.15. Consultar Cursos

Tabla A.15. Prueba de aceptación # 25

Caso de prueba de aceptación	
Código: HU25_P25	Historia de usuario: 25
Nombre: Consultar Cursos.	
Descripción: Prueba para la funcionalidad Consultar Cursos.	
Condiciones de ejecución: El VIP1 es el encargado de publicar las solicitudes de cursos luego de revisarlas que a la vez que la publica aparece en la portada del sistema para informar a los usuarios del sistema. El sistema le permite a los usuarios matricularle si lo desean en los cursos publicas siempre y cuando haya capacidad y no esté ya matriculado en el curso.	
Pasos de ejecución: En la portada del sistema le aparece la opción <i>Cursos</i> . El usuario accede a la misma y se informa mediante dicha información publicada los cursos que se están ofertando o se van a ofertar en la facultad. Los usuarios podrán acceder a todos los documentos que se han publicado en el sistema.	
Resultados esperados: El usuario consulta los cursos en el sistema.	

A.16. Consultar Anuncios

Tabla A.16. Prueba de aceptación # 26

Caso de prueba de aceptación	
Código: HU26_P26	Historia de usuario: 26
Nombre: Consultar Anuncios.	
Descripción: Prueba para la funcionalidad Consultar Anuncios.	
Condiciones de ejecución: El VIP1 es el encargado de publicar las solicitudes de anuncios luego de revisarlas que a la vez que la publica aparece en la portada del sistema para informar a los usuarios del sistema. El sistema le permite a los usuarios informarse sobre eventos, concursos que se van a desarrollar.	
Pasos de ejecución: En la portada del sistema le aparece la opción <i>Anuncios</i> . El usuario accede a la misma y se informa mediante dicha información publicada. Los usuarios podrán acceder a todos los anuncios que se han publicado en el sistema.	
Resultados esperados: El usuario consulta los anuncios en el sistema.	

A.17. Consultar Evidencias de Usuario

Tabla A.17. Prueba de aceptación # 27

Caso de prueba de aceptación	
Código: HU27_P27	Historia de usuario: 27
Nombre: Consultar Evidencias de Usuario.	
Descripción: Prueba para la funcionalidad Consultar Evidencias de Usuario.	
Condiciones de ejecución: El usuario tiene que autenticarse correctamente en el sistema.	
Pasos de ejecución: El usuario puede acceder a consultar las evidencias de los usuarios del sistema en la portada en el botón Ver más , en sus solicitudes de evidencias y en el menú superior en Servicios . Puede filtrar a la hora de buscar por usuarios, áreas y fecha, y obtener las evidencias de los usuarios deseados.	
Resultados esperados: El usuario consulta las evidencias de los usuarios del sistema.	

B.1. LoginController

Tabla B.1. Tarjeta CRC # 11

Tarjeta CRC	
Clase: LoginController	
Responsabilidad	Colaboración
Esta clase es la encargada de todo lo que trata con el logueo y acceso al sistema, además de que tiene la responsabilidad de crear los roles que van a ser asignados a los usuarios que naveguen en el sistema si dichos roles no existen en la base de datos. También verifica que haya conexión con el servidor de la UCI para realizar las peticiones correspondientes al mismo.	Rol Usuario AuntenticacionTypeType

B.2. ExportarpdfController

Tabla B.2. Tarjeta CRC # 12

Tarjeta CRC	
Clase: ExportarpdfController	
Responsabilidad	Colaboración

Continúa en la próxima página

Tabla B.2. Continuación de la página anterior

Esta clase tiene la responsabilidad de permitir exportar a PDF informes del estado de cumplimiento de los indicadores en la Facultad 2.	Controller
---	------------

B.3. DescargasController

Tabla B.3. Tarjeta CRC # 13

Tarjeta CRC	
Clase: DescargasController	
Responsabilidad	Colaboración
Esta clase tiene la responsabilidad de gestionar los archivos que son o pueden ser de interés al claustro post-graduado perteneciente a la Facultad 2.	Descargas DescargasType

B.4. DefaultController

Tabla B.4. Tarjeta CRC # 14

Tarjeta CRC	
Clase: DefaultController	
Responsabilidad	Colaboración
Es la clase encargada de mostrar las informaciones pertinentes y de interés en la portada del sistema para que el usuario puede acceder a dicha información sin mucho manejo de flujos dentro del sistema.	Controller

B.5. GraficosController

Tabla B.5. Tarjeta CRC # 15

Tarjeta CRC	
Clase: GraficosController	
Responsabilidad	Colaboración

Continúa en la próxima página

Tabla B.5. Continuación de la página anterior

En esta clase se maneja todo lo que tiene que ver con la visualización de información en gráficos.	Controller
--	------------

B.6. AnuncioController

Tabla B.6. Tarjeta CRC # 16

Tarjeta CRC	
Clase: AnuncioController	
Responsabilidad	Colaboración
Esta clase se encarga de administrar las solicitudes de los usuarios de publicar un anuncio. Anuncios que son de interés al personal que accede al sistema.	Anuncio AnuncioType BuscaranunciosType

B.7. CursoController

Tabla B.7. Tarjeta CRC # 17

Tarjeta CRC	
Clase: CursoController	
Responsabilidad	Colaboración
Esta clase como su nombre lo indica tiene la responsabilidad de gestionar el tema referente a las solicitudes de aquellos profesores que desean impartir un curso y han cumplido con los requisitos para impartirlo, en esta clase de administran dichas solicitudes.	Curso CursoType BuscarcursosType

B.8. InformeController

Tabla B.8. Tarjeta CRC # 18

Tarjeta CRC	
Clase: InformeController	
Responsabilidad	Colaboración

Continúa en la próxima página

Tabla B.8. Continuación de la página anterior

<p>Esta clase gestiona la información referente a todo lo que tiene que ver con generar informes a partir informaciones ya existentes dígame los indicadores, la asignación de planes de indicadores a las áreas y la gestión de evidencias en el sistema.</p>	<p>Controller</p>
--	-------------------

B.9. CategoriabajadorController

Tabla B.9. Tarjeta CRC # 19

Tarjeta CRC	
<p>Clase: CategoriabajadorController</p>	
Responsabilidad	Colaboración
<p>Esta clase gestiona la información referente a las categorías docentes que le van hacer asignadas a los profesores que acceden al sistema. Pues de la gestión correcta de esta clase dependen otras como CursoController para a la hora de realizar una solicitud de impartir un curso dicho profesor debe de cumplir que su categoría docente es <i>Asistente</i> o superior.</p>	<p>Categoriabajador CategoriabajadorType</p>